UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Master Degree in Physics

Final Dissertation

Personalization of Vision-language Models and the Multi-Concept Challenge

Thesis supervisor

Prof. Pietro Zanuttigh

Thesis co-supervisor

Prof. Elisa Ricci

Candidate

Gloria Isotton

Academic Year 2024/2025

To my family, for their unwavering support,
and to Giovanni, for walking this path with me.

Contents

Contents

Abstract

Recent advances in vision and language models have led to significant improvements across a wide range of applications, substantially transforming human–machine interaction. Despite these achievements, current models continue to face limitations in the domain of personalization, particularly when required to generate outputs grounded in user-specific concepts. This work investigates the problem of personalization in vision and language models, with a specific emphasis on evaluating model behaviour in scenarios where multiple user-defined concepts are simultaneously present within a single image.
To this end, a novel dataset, MP-Bench, is introduced, designed to support evaluation in both single- and multi-concept personalization settings. The dataset is constructed using personalized generative models, demonstrating a scalable and flexible approach to dataset creation in this context. Furthermore, an extensive evaluation of existing models is conducted on both the proposed dataset and publicly available benchmarks, providing a comprehensive analysis of current performance in the personalization task.

Acknowledgments

I would like to thank my supervisor, Davide Talon, for his valuable advice and continuous support throughout this work. I am also grateful to the entire research group at the Deep Visual Learning Laboratory in Trento, as well as to Professor Elisa Ricci, for their guidance, insights, and collaborative spirit.
I would like to express my sincere gratitude to my internal thesis supervisor, Pietro Zanuttigh, for his support and consistently helpful advice throughout this journey.
A heartfelt thank you goes to my partner, Giovanni: you know how challenging this path has been, because you walked it too, and your presence has made all the difference.
I would also like to warmly thank my family, my parents and my sister, for their constant encouragement and unconditional support.
Finally, I am grateful to the professors of the Master's program in Physics of Data who have often been a source of inspiration, and to the University of Padua for providing welcoming and useful study spaces during these years.

Gloria Isotton

Organization of the Work

The structure of this work is organized as follows. Chapter 1 introduces the general problem addressed in this work, outlines its objectives, and presents the questions this work aims to investigate. A brief overview of the adopted methodology is provided, together with a concise summary of relevant contributions from the existing literature. Chapter 2 presents a review of related work, offering an overview of existing research on the topic of personalization in vision-language models.
Chapter 3 details the construction of the proposed dataset, MP-Bench. It begins with a formal definition of the problem, followed by a review of prior work in text-to-image generation and, more specifically, on personalized generative models. The chapter includes a technical description of the dataset creation pipeline, which leverages MS-Diffusion as the underlying personalized generative model.
Chapter 4 describes the experimental setup, including the configuration of the experiments and the evaluation metrics employed. Chapter 5 presents an analysis of the obtained results, while Chapter 6 concludes the work by reflecting on the findings, discussing current limitations, and outlining possible directions for future research.

Chapter 1

Introduction

1.1 Defining Personalization in vision-language models

In this work, I will present the results of my research on the topic of Personalization in computer vision. Given that the term can encompass multiple interpretations, it is important to establish a precise definition from the outset. Here, personalization is defined as the process of adapting models to recognize specific object instances and to generate responses that are tailored to those user-defined concepts.

1.2 Motivations and real-world applications of Personalization

In recent years, large language models (LLMs) and vision language models (VLMs) have achieved remarkable progress, demonstrating their effectiveness across a wide range of applications and significantly transforming the way humans interact with machines. Despite these advances, current models continue to exhibit notable limitations in the domain of personalization, particularly when required to generate responses grounded in user-specific concepts. For example, when presented with images that include a custom object or concept, such as a user's personal pet or a unique product prototype, current models often struggle to accurately distinguish the specific instance from the general concept. This challenge can be attributed to two primary factors. First, personalized concepts are typically absent from large-scale training datasets. Consequently, models lack prior knowledge of the visual identity or user-assigned name associated with such concepts. Second, constructing large-scale datasets for personalization is inherently difficult, as the task requires multiple, diverse visual instances corresponding to the same concept. Such data are rarely available in natural settings and often need to be artificially synthesized. These limitations contribute to the models' tendency to conflate specific instances with general categories, thereby hindering their ability to produce accurate personalized outputs.
This limitation poses a significant barrier to the integration of VLMs into real-world scenarios, where context-aware, personalized responses are often crucial. This has led to the growing popularity of the personalization task, which demonstrates significant potential across a broad range of applications, from facilitating daily tasks through personalized virtual assistants, to addressing more impactful needs, such as enhancing accessibility for blind or low vision individuals by enabling the identification and retrieval of personal items [1].

1.3 Overview of Personalization in the Literature

As mentioned above, the term *personalization* is used in the literature to refer to related, though slightly different, tasks. For the sake of clarity, we can group these into three main categories. The first definition is rather specific to a single paper [2], which uses the term personalization to describe the exploitation of interrelated properties within a dataset and the leveraging of intra-user consistency

to improve segmentation. However, this definition appears only in that work, is highly specific to its context, and relies on a specialized dataset. For these reasons, it will not be consider further in this analysis.

The second definition appears in a set of publications [3, 4, 5], where personalization involves using a user-specific visual concept, typically represented by an image, as a reference for recognizing and segmenting that concept in other images. At this stage, the notion of personalization begins to emerge, although the process remains entirely visual. No linguistic label is associated with the user-defined concept, limiting its semantic integration and distinguishing it from approaches that involve language grounding.

PerSAM [3] introduces a zero-shot approach to personalized, image-prompted segmentation. Given a test image and a reference image, both containing the target concept, PerSAM leverages a Location Confidence Map combined with a novel Target-Guided Attention mechanism to provide SAM with both positive and negative location priors of the target object in the test image.

IPSeg [4], on the other hand, exploits features extracted from Stable Diffusion and DINO to obtain a robust representation of both the test and reference images. Through a feature interaction module, the model generates a small set of positive and negative points on the test image, which are then used as prompts for SAM.

The PDM approach [5] utilizes pre-trained diffusion features to segment a specific reference object in a new scene, with a particular focus on challenging scenarios involving visual distractors, that is, objects of the same class present within the test image.

The spirit of the personalization task, namely the ability to adapt models to recognize specific object instances and distinguish them from more general concepts, is indeed present in these works. However, these models operate primarily as image-prompted segmentation pipelines, with limited personalization beyond visual conditioning.

The third and most relevant definition [6, 7, 8], which this work focuses on, refers to the personalization of user-specific concepts by augmenting the input vocabulary of a pretrained model with new word embeddings that represent those personalized concepts.

PerVL [6], introduced in 2022, was the first approach to establish a framework for personalizing pretrained VLMs. It requires only a few positive image examples, without needing any negative samples. The core idea behind PerVL is to enable the expansion of the VLM's input vocabulary by learning to associate new visual concepts, defined by a small set of example images, with corresponding word embeddings. To achieve this, the authors propose an inversion mapping module, denoted as f_θ, which is trained on large-scale, non-personalized datasets. This module maps CLIP image-space output embeddings (i.e., representations of a set of images in CLIP's output space) to corresponding word embeddings in CLIP's input space. During the personalization phase, the user provides a small set of images representing the target concept. These images are encoded in CLIP's image embedding space and then passed through the inversion module f_θ to generate an initial word embedding corresponding to the personalized concept. This embedding can subsequently be refined through fine-tuning, enabling the model to more accurately capture the semantics of the user-specific concept and incorporate it into the broader VLM vocabulary. This approach allows for efficient personalization of VLMs without retraining the entire model.

MyVLM [7] introduces a different approach to personalization by augmenting VLMs with external concept heads, a module that allow the model to detect whether a specific target concept is present in a given image. Once the concept is recognized, the system learns a new embedding for it within the intermediate feature space of the VLM. This embedding is then used to guide the language generation process, helping the model naturally incorporate the personalized concept into its output.

Yo'LLaVA [8] introduces a token-based method for personalization by adding a set of learnable input tokens to the VLM. Specifically, it introduces one special identifier token, `<sks>`, and k latent tokens. The special token serves as a symbolic reference to the personalized concept, allowing both the user and the model to refer to the same concept explicitly. The k latent tokens are designed to capture the visual characteristics associated with `<sks>`. Training is limited to the $k + 1$ newly introduced input tokens and the final classifier head matrix W associated with the `<sks>` token. The model is

trained using a hard negative mining strategy, which encourages themodel to learn fine-grained visual distinctions and to differentiate the target concept from other visually similar objects belonging to the same category.
These studies primarily focus on single-concept personalization, where the model is required to handle only one user-defined concept at a time. However, this line of work does not fully address the more complex scenario in which multiple personalized concepts must be integrated within a single image. This limitation is the focus of the next section, which discusses a small but growing body of recent work that begins to explore multi-concept personalization.

1.4 Personalizing multiple concepts in a single scene

While the field of VLMs personalization has seen important developments in enabling models to recognize and respond to user-specific concepts, the majority of existing approaches focus primarily on single-concept personalization. In this setting, models are adapted to understand and generate responses conditioned on a single, user-provided linguistic reference, for example, a particular pet, object, or person. Although effective in constrained scenarios, this focus limits the applicability of such systems in real-world environments, where users naturally refer to multiple personalized entities within a single interaction. In practice, users may wish to describe or query scenes involving several distinct personalized concepts simultaneously.
The presence of multiple user-specific concepts in a single image or prompt adds an extra layer of complexity. Models must be able to recognize each concept individually, while also reasoning about their spatial and semantic relationships. Achieving this without confusion between concepts or degradation of output quality remains difficult. In this work, I address this challenge, framing multi-concept personalization as a central problem for the development of more reliable and context-aware vision-language systems.
The task of handling multiple personalized concepts within a single image is relatively new and only explored in a handful of recent studies [9, 10].
A significant limitation in this area is the lack of dedicated datasets tailored to the multi-concept setting. An ideal dataset for multi-concept personalization should include diverse scenarios featuring not only individual user-specific subjects but also their combinations, ranging from pairs up to groups of five or six concepts within the same scene. This would allow models to learn both isolated and co-occurring representations of user-defined entities.
Recent works addressing multi-concept personalization have proposed original strategies to overcome this limitation. One approach involves extracting relevant frames from videos to capture natural co-occurrences of multiple concepts. Another promising direction, explored in this work, leverages generative models to synthesize personalized datasets that include varied and controllable multi-concept compositions. This synthetic data generation offers a flexible solution to support training and evaluation in this emerging task.

1.5 Main Contributions of the Thesis

This thesis makes two main contributions to the field of personalized vision-language modeling. First, a novel synthetic dataset was developed leveraging the MS-Diffusion model [11] for evaluating personalized VLMs. The dataset, referred to as MP-Bench, is divided into two splits: one derived from the publicly available Yo'LLaVA dataset [8], and the other from the MS-Bench dataset [11]. Each split includes a collection of single personalized concept images, categorized into training and testing sets, as well as images featuring multiple personalized concepts within the same scene. All the images depict a variety of scenarios, ranging from outdoor environments (such as parks, beaches, and urban areas) to indoor settings, ensuring diversity in background and context. The personalized concepts span a wide range of categories, including pets, humans, clothing items, and everyday objects. The final dataset comprises a total of 315 multi-subject images (each containing 2 or 3 target concepts), and includes

80 distinct personalized concepts overall.
In addition to the dataset creation, this thesis provides a thorough experimental evaluation demonstrating that the synthetic data can effectively be used to test personalized vision-language models without significant performance loss compared to real, manually curated datasets. This finding supports the use of generative approaches, such as MS-Diffusion, as practical and scalable alternatives to traditional data collection methods in personalization research.

Furthermore, this thesis presents a comprehensive benchmark of several state-of-the-art personalization models, evaluated on both publicly available datasets and the newly proposed synthetic dataset. This systematic evaluation not only validates the utility of the synthetic data for personalization tasks but also offers detailed insights into the strengths and limitations of each method, with particular attention to the trade-offs between downstream task performance, training time (where applicable), and scalability.

Collectively, these contributions aim to support the development of robust, efficient, and scalable personalization strategies for vision-language systems.

Chapter 2

A survey of Personalization in vision-language models

In recent years, the growing adoption of VLMs has shown remarkable potential across a wide range of tasks due to their broad and generalizable knowledge. These models are typically trained on massive datasets composed of diverse images and text, enabling them to perform well across various domains without task-specific tuning. However, this generality also introduces a critical limitation: human users often communicate using highly specific, personalized terms and references, such as names of people, pets, objects, or uncommon concepts that fall outside the scope of the models' original training data. This creates a gap between the model's general capabilities and the user's individualized needs.
VLMs Personalization task aims to bridge this gap by adapting the model to recognize and understand user-specific concepts, meaning the novel categories that were not part of its training distribution. The core challenge lies in enabling the model to learn and generalize these new classes from only a handful of labeled examples, while ensuring that its broad pre-trained knowledge remains unaffected (i.e., avoiding catastrophic forgetting).
Research in this area is currently very active, and the literature offers a range of approaches to address the Personalization problem. Broadly speaking, existing methods can be grouped into two main categories (Table 2.1): those that rely on additional training, and training-free approaches. Each of these paradigms comes with its own advantages and trade-offs in terms of flexibility and overall performances.

	Images		**Requirements**			**Support**	
	Positive	**Negative**	**Caption**	**Description**	**Training**	**Real-time edit**	**Text-only QA**
MyVLM [7]	10-15	150	Yes	No	Yes	×	×
Yo'LLaVA [8]	5-10	200	No	No	Yes	×	✓
MC-llava [10]	10	150	No	No	Yes	×	✓
RAP [12]	1	-	No	Yes	No	✓	✓

Table 2.1: Comparison of personalization methods for multimodal models.

2.1 Training-Based Personalization Methods

2.1.1 MyVLM: Personalizing VLMs for User-Specific Queries

MyVLM [7] is a training-based personalization methodology designed to personalize Vision-Language Models.
The core of this approach lies in the integration of external concept heads, which are trained to identify user-specific concepts within a given image. The signal from these heads is used to introduce specific,

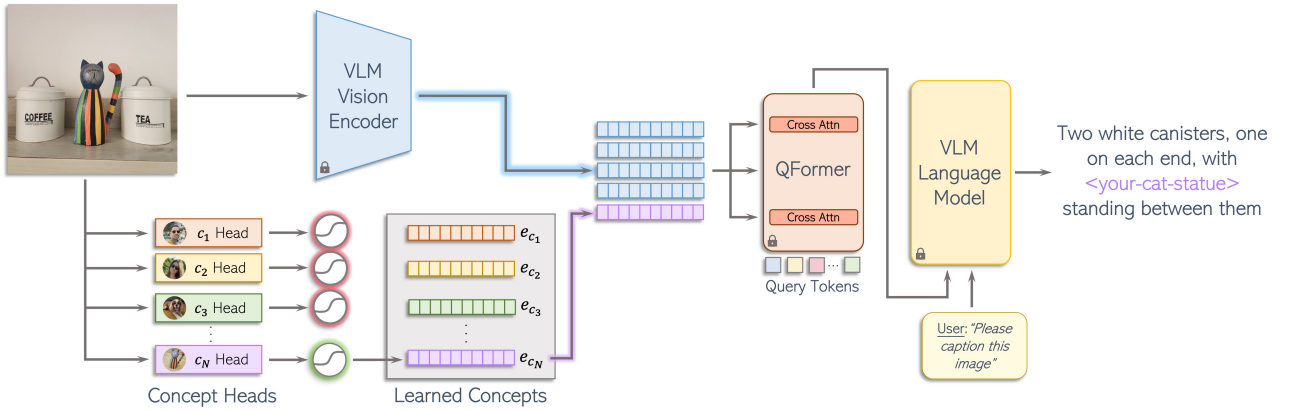


Figure 2.1: MyVLM overview, applied over BLIP-2. The application of MyVLM to the BLIP-2 model introduces a targeted modification to its core data flow, enabling personalization without altering foundational weights. The process begins by feeding an input image simultaneously to the frozen vision encoder and a bank of custom-trained concept heads. While the encoder extracts a broad visual representation, the concept heads are tasked with detecting the presence of specific, user-defined entities. If a known concept is identified, its unique embedding is appended to the vision encoder's output features. This creates an augmented feature representation that contains both global visual information and specific conceptual cues. This enriched representation is then passed to the Q-Former module. The Q-Former's cross-attention layers distill the most salient information from this combined input. Finally, the resulting output from the Q-Former, along with a textual instruction, conditions the frozen LLM to generate a response. *Figure adapted from Alauf et al. (2024).*

learnable concept vectors alongside the output of the vision encoder. These vectors are tasked with guiding the Language Model's (LM) response generation, ensuring that the personalized term is incorporated in a way that is both contextually accurate and semantically aligned with the input image. The training of these vectors is parameter-efficient, requiring only a small set of 3-5 images depicting the concept, each accompanied by a caption containing the personalized word. Architecturally, MyVLM is built upon foundational models and it is trained without altering the original model weights.

The MyVLM personalization framework has been developed primarily on top of two prominent families of Vision-Language Models: BLIP-2 and LLaVA. These models represent distinct architectural paradigms within the multimodal learning landscape.

BLIP-2 model, introduced by *Li et al.* [13], consists of three main components: a pre-trained vision encoder, typically a Vision Transformer (ViT-L/14) [14], which remains frozen throughout training; a pre-trained large language model [15], also kept frozen; and a trainable Querying Transformer (Q-Former), which serves as the intermediary between vision and language. The Q-Former operates by introducing 32 learnable query tokens (each of dimension $d = 768$), which interact with the visual features via a combination of self-attention, cross-attention, and feed-forward layers. Through cross-attention with the frozen vision encoder outputs, the Q-Former distills salient visual information into a fixed-length representation suitable for the language model. This design allows for flexible and efficient visual grounding without updating the core components.

LLaVA [16] follows an end-to-end architecture that emphasizes simplicity and direct modality alignment. The model integrates a CLIP-based vision encoder (ViT-L/14) [17] with the Vicuna large language model [18]. Rather than employing a dedicated bridging module like BLIP-2, LLaVA uses a single linear projection layer to map visual features from the encoder into the token embedding space of the language model. This enables the language model to jointly process visual and textual inputs in a unified sequence.

MyVLM: Method

This section provides a detailed explanation of the MyVLM architecture, built upon the BLIP-2 backbone, as illustrated in Figure 2.1. The model is designed to enable a VLM to recognize and reason about novel concepts using only a few (approximately 3–5) reference images and their corresponding captions, which include a unique concept identifier S^*. The goal is to equip the VLM with the capability to respond to queries about new images that depict the target concept.
The MyVLM framework is composed of two primary components: a concept head module responsible for identifying the presence of the target concept in the input image, and a concept embedding mechanism that facilitates communication of concept-specific information to the language model.
Concept head module. Empirical observations indicate that the frozen vision encoder in BLIP-2 lacks sufficient expressiveness to reliably distinguish the target concept from visually similar ones. To address this, a set of external concept heads, each dedicated to recognizing a specific, user-defined concept was introduced. These concept heads function independently of the core VLM and consist of simple linear classifiers trained on embeddings extracted from a pre-trained CLIP model. For the personalization of human subjects, an additional face recognition module is incorporated to the model, leveraging a pre-trained face recognition network to more accurately identify instances of the personalized subject S^* within an image.
Concept embedding module. The second stage of the pipeline focuses on enabling the VLM to generate responses that incorporate the newly learned concept. For each concept, the model learns a dedicated concept embedding vector e^*, representing the concept within the intermediate feature space of the VLM. This embedding is optimized to steer the language model toward producing captions that explicitly mention the designated concept identifier.
The optimization process involves a small set of training images depicting the concept in diverse contexts, each accompanied by a target caption containing the identifier, which is chosen to be a real but uncommon word to avoid semantic collisions. The goal is to train e^* such that, when it is appended to the image features extracted by the frozen vision encoder, the downstream VLM will generate the desired personalized caption.
Specifically, the embedding e^* is appended to the image features extracted by the frozen vision encoder and passed to the Q-Former via cross-attention layers. The Q-Former output is subsequently fed into the frozen language model, which generates the predicted caption.
To ensure that the generated caption matches the target caption, the concept embedding is optimized using the cross-entropy loss between the generated and target sequences. This loss function penalizes deviations from the ground-truth caption at each token position, guiding the model to produce text that closely aligns with the desired output. Formally, the optimization objective is defined as:

$$e^* = \arg\min_e \sum_{i=1}^{N} \mathcal{L}_{CE}(t_i, o(I_i, e)),$$

where N is the number of training samples, t_i is the target caption for the i-th image I_i, $o(I_i, e)$ is the caption generated by the model when conditioned on the image I_i and the concept embedding e and $\mathcal{L}_{CE}$ denotes the standard cross-entropy loss computed over the token sequence.
It should be noted that directly appending the concept embedding to the image features can lead to degraded caption quality, often resulting in unnatural outputs. To mitigate this, two strategies are implemented: the first consists of normalizing the concept embedding keys and values to match the average norm of the original keys and values prior to cross-attention computation in the Q-Former, and the second consists of applying an L2 regularization term to the attention probabilities assigned to the concept embedding across all 32 Q-Former query tokens.
These strategies are specifically tailored to the BLIP-2 architecture. When adapting the model to LLaVA, which lacks a cross-attention mechanism, the concept embedding is rescaled to match the norm of the $[CLS]$ token output from the vision encoder, omitting key value normalization.

MyVLM: Training details

The training of each concept head involves distinguishing between four positive samples (i.e., images containing the target concept) and 150 negative samples, which are images of similar-looking objects from the same broad category sourced from the internet. For example, when training a concept head to recognize a specific dog, the negative samples consist of arbitrary images of other dogs. Training is performed for 500 steps using a standard cross-entropy loss function, with a batch size of 16 and the AdamW optimizer.
In contrast, training of the concept embedding is conducted as follows: when applied to the BLIP-2 architecture, 75 optimization steps for object concepts and 100 steps for personalized human subjects are performed. For LLaVA, 100 steps are employed for both categories.

2.1.2 Yo'LLaVA : Pioneering Personalized Training in MLMs

Yo'llava [8] introduces a novel training-based personalized multimodal language model (MLM) built upon the state-of-the-art LLaVA framework [16]. The core idea is to personalize the model using only 5 to 10 images per concept, along with a set of negative samples. From this limited data, Yo'llava learns to embed each concept into a compact set of special tokens. These tokens are then used during inference to condition the model, enabling it to understand and answer questions about the personalized concept when prompted.
From a technical perspective, Yo'llava achieves personalization by freezing nearly all of the pre-trained weights of the underlying MLM and introducing a small number of additional trainable parameters into selected layers. The approach follows the soft prompt tuning paradigm, where a set of learnable input tokens is used to inject personalized information into the model. These consist of a single special token $\langle \texttt{sks} \rangle$ and a sequence of latent tokens $\langle \texttt{token}_1 \rangle, \langle \texttt{token}_2 \rangle, ..., \langle \texttt{token}_k \rangle$. Trainable output weights are associated with the $\langle \texttt{sks} \rangle$ and latent token to refine the model's responses. Crucially, this personalization is achieved without compromising the general-purpose capabilities of the original model.
Once trained, Yo'llava demonstrates several key capabilities. It is able to identify the subject within new images during the testing phase, such as determining whether a specific entity, denoted as $\langle \texttt{sks} \rangle$, is present in a given photograph. Beyond recognition, Yo'llava supports visual question answering focused on the subject; for instance, it can respond to inquiries about the location of $\langle \texttt{sks} \rangle$ in a novel image. Additionally, the system enables text-only discussions concerning the subject without needing any accompanying images during the test phase. This allows it to answer questions about intrinsic attributes of $\langle \texttt{sks} \rangle$, including characteristics like color, shape, and other inherent properties.

Yo'LLaVA : Method

This section details the methodology of the Yo'LLaVA framework, as illustrated in Figure 2.2. The core objective of Yo'LLaVA is to personalize a MLM, specifically LLaVA [16], by grounding it to a novel subject, using only a small set of images (e.g., five images of a user's dog, referred to as $\langle \texttt{sks} \rangle$). These images are provided without any textual annotations or captions.
Before introducing the technical aspects of the method, an example is presented to highlight why latent, learnable tokens constitute a more practical and effective alternative to manually written descriptions for representing user-specific subjects.
Let us consider the case of a user who wishes to determine whether a specific object, such as his dog, referred to as $\langle$snow$\rangle$, appears in an image. A natural strategy might involve providing a textual description (e.g., "$\langle$*snow*$\rangle$ *is a medium-sized dog with brown fur...*"). However, crafting such prompts manually is often challenging and prone to imprecision. Accurately describing the visual appearance of a subject in words can be lengthy and, in many cases, insufficient. Subtle visual cues, such as the particularities of a person's facial features, expressions, or hairstyle, are frequently too nuanced to be captured adequately through text alone.
To address these limitations, Yo'LLaVA adopts a learnable prompting approach. Rather than relying on hand-written descriptions, it utilizes soft prompts: continuous, learned token embeddings that en-

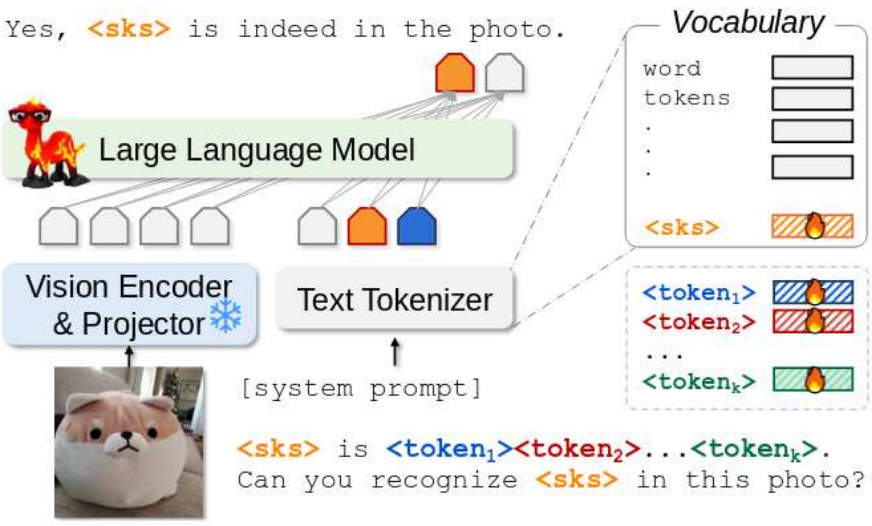


Figure 2.2: Yo'LLaVA Training pipeline. In this approach almost all the pre-trained weights of the base MLM model are freezed. A small set of trainable parameters is introduced in the form of learnable input tokens: $\langle \texttt{sks} \rangle$ is $\langle \texttt{token}_1 \rangle, \langle \texttt{token}_2 \rangle, ..., \langle \texttt{token}_k \rangle$. The model is personalized by training these tokens along with associated output weights using a small number (5 to 10) of images per concept, combined with negative samples. *Figure adapted from Nguyen et al. (2024).*

code a personalized representation of the subject. The process involves introducing a small number of learnable vectors, the prompt tokens, and their associated output weights. Such embeddings capture the essential visual features derived from the provided exemplar images. This mechanism allows for the integration of new subject-specific knowledge into the MLM without modifying its general-purpose parameters or retraining the entire model.

The introduction of a new personalized subject, denoted as $\langle \texttt{sks} \rangle$, is accompanied by a set of positive reference images $I_1, \ldots, I_n$ depicting that subject. During training, a personalized soft prompt is defined in the form $\langle \texttt{sks} \rangle$ is $\langle \texttt{token}_1 \rangle, \langle \texttt{token}_2 \rangle, ..., \langle \texttt{token}_k \rangle$, where the new special token $\langle \texttt{sks} \rangle$ serves as a unique identifier to allow both the user and the model to refer to the subject in questions and answers. The remaining tokens $\{\langle \texttt{token}_i \rangle\}_{i=1}^{k}$ are soft tokens learned specifically to embed the visual details of the subject, extracted directly from the provided reference images.

To accommodate this new subject, the final classifier head matrix W of the language model is expanded from dimensions $C \times N$ to $C \times (N+1)$, where C denotes the hidden feature dimension and N is the original vocabulary size. Within the Yo'llava framework, the trainable parameters are restricted to the set $\theta = \{\langle \texttt{sks} \rangle, \langle \texttt{token}_1 \rangle, \ldots, \langle \texttt{token}_k \rangle, W(:, N+1)\}$, meaning that only the newly introduced input tokens and the classifier weights associated with the $\langle \texttt{sks} \rangle$ token are updated during training. All other components of the pre-trained LLaVA model, including the vision encoder, vision projector, and core language model, remain frozen.

To enhance the model's understanding and recognition of the new visual concept, conversational training data is constructed as triplets (I_i, X_i^q, X_i^a), each consisting of an input image I_i, a corresponding question X_i^q, and an answer X_i^a. The training process employs a standard masked language modeling loss, computing the probability of the target response X^a for a conversation of length L according to the formula

$$p(X^a \mid I_i) = \prod_{j=1}^{L} p_\theta(x_j \mid I_i, X^a_{<j}),$$

where θ denotes the trainable parameters and $X^a_{<j}$ represents the instruction and answer tokens preceding the current token x_j.

To effectively train the model to recognize and reason about a user-specific subject, such as $\langle \texttt{sks} \rangle$, four set of conversational training data triplets are constructed. The types of training data used include:

- *Positive Recognition triplets*: These examples use reference images that contain the subject ⟨sks⟩. The associated queries ask whether ⟨sks⟩ is present in the image, and the ground truth answers are always affirmative, sampled from a pool of positive response templates. While such examples are essential for learning the subject's identity, training exclusively on them can lead to shortcut learning, where the model tends to answer *"Yes"* regardless of the actual content of the image.

- *Soft Negative Recognition Triplets*: To counteract this bias, additional triplets are generated using randomly sampled images from LAION [19] a large-scale open-source collection of 5.85 billion CLIP-filtered image-text pairs scraped from the web. These images do not contain ⟨sks⟩. The queries maintain the same form as in the positive case (i.e., asking whether ⟨sks⟩ is present), but the ground truth answers are always negative, selected from a set of negative response templates. This mixture of positive and soft negative examples encourages the model to learn more discriminative visual features and develop a better understanding of the subject's visual attributes.

- *Hard Negative Recognition Triplets*: To further improve discriminative ability and prevent over-generalization, hard negatives are introduced. These consist of images that contain visually similar, but not identical, objects relative to ⟨sks⟩. For instance, if ⟨sks⟩ is a user's dog, hard negatives could include other dogs with similar color and size. Such hard negative images are retrieved from the LAION dataset [19] using top-k CLIP embedding similarity to the reference images of the target subject. These examples force the model to focus on fine-grained visual distinctions.

- *Text-Only Subject Reasoning Pairs*: In addition to recognition tasks, text-only conversations are incorporated to help the model reason about the subject itself. These involve image-grounded template questions that probe the subject's visual characteristics, such as *"What is the hair color of this person?"* or *"What is this object made of?"*. For each reference image I_i, the LLaVA model is used to generate plausible answers based on the image content. These text-based examples complement the recognition tasks. It's important to note that, in this specific case, the Yo'LLaVA model is trained with question-answer pairs rather than full triplets. The image reference is deliberately excluded during this training phase. This is because providing the image would often give the model enough information to answer the question, even without truly understanding the unique visual attributes of the personalized subject. For example, if shown a photo of a stuffed animal ⟨sks⟩ and asked *"What color is it?"*, LLaVA could likely answer correctly based on the image alone, without needing to process or comprehend the visual characteristics associated with ⟨sks⟩ itself.

Yo'LLaVA : Training details

The final Yo'LLaVA model was trained on a custom dataset specifically curated for personalized vision-and-language tasks. This dataset comprises 40 diverse user-defined subjects, systematically categorized into five groups: 10 individuals, 5 pets, 5 landmarks, 15 objects, and 5 fictional characters. Each subject is represented by 10 to 20 reference images and is assigned to either a training or testing split to ensure proper evaluation.
The default training configuration utilized five reference images per subject and employed a soft prompt length of $k = 16$, corresponding to 16 learnable tokens per subject.
The model was initialized from LLaVA-1.5-13B and trained using the AdamW optimizer. To improve robustness and discriminative ability, each subject's training set was augmented with approximately 200 negative images: 100 hard negatives obtained via image retrieval based on CLIP embedding similarity, and 100 easy negatives sampled randomly. Training was performed for up to 15 epochs per subject, and the best model checkpoint was selected according to recognition accuracy on the training set.
The performance of the trained model was systematically evaluated on both recognition and visual

question answering (VQA) tasks, and compared against several baselines, including MyVLM [7], as well as general-purpose models such as GPT-4V [20], LLaVA [16], and LLaVA augmented with manually written personalized descriptions.

Yo'LLaVA-M

Yo'LLaVA-M is a variant of the original Yo'LLaVA model [8] (here referred to as Yo'LLaVA-S) introduced in the MC-LLaVA paper [10] designed to support the same downstream tasks, namely, recognition and captioning, while also enabling the simultaneous handling of multiple target concepts at inference time.
In the original Yo'LLaVA-S model, performing a recognition task for a specific target concept `<sks>` requires loading the pre-trained concept embeddings and the associated language modeling head. This architecture is inherently limited to handling a single target concept at a time.
Yo'LLaVA-M maintains the general architecture and training strategy of Yo'LLaVA-S but introduces specific modifications to allow for multi-concept reasoning. Based on the description provided in the MC-LLaVA paper, where the multi-concept extension was first introduced, Yo'LLaVA-M is designed to merge separately trained concept embeddings and augment the classification head parameters so that the model can respond to queries involving multiple learned concepts simultaneously. However, due to the absence of a detailed implementation and publicly available code, the version of Yo'LLaVA-M used in this work is derived from an independent interpretation of the original description.
Two primary modifications with respect to the original Yo'LLaVA-S model are introduced in the implementation adopted here:

- Assignment of Auxiliary Token Names During Training: in Yo'LLaVA-S, auxiliary tokens representing the special concept token `<sks>` could share the same names across different concepts (e.g., numbered from 1 to 15) since only one concept's tokens were loaded at a time during inference. In contrast, Yo'LLaVA-M requires simultaneous loading of multiple token sets. To avoid naming conflicts, a token offset is introduced for each concept, and separate embeddings are trained for each.
- Multi-Concept Inference: at inference time, the model is modified to load all relevant pre-trained embeddings and LM heads corresponding to the desired concepts simultaneously. This enables the system to perform recognition and captioning tasks that involve multiple personalized concepts within the same query or input.

2.1.3 MC-LLaVA : Multi-Concept Personalization for Vision-Language Models

MC-LLaVA [10] is a novel training-based approach specifically designed for the multi-concept personalization paradigm and derived from the Yo'LLaVA [8] architecture. Unlike previous personalization methods that focus on a single concept at a time, MC-LLaVA introduces a multi-concept instruction tuning strategy that enables the integration of multiple concepts within a single training step. This model was proposed to address two key limitations of the earlier personalized large multimodal model, Yo'LLaVA.
The first limitation of Yo'LLaVA is that it trains each concept independently, and combining parameters for different concepts results in performance degradation due to concept confusion.
The second issue is its heavy reliance on high-quality negative samples. MC-LLaVA mitigates both problems by simultaneously considering multiple concepts during training, rather than treating them independently.

MC-LLaVA : Method

The method's pipeline, as shown in Figure 2.3, comprises three main components: multi-concept instruction tuning, personalized textual prompts for multiple concepts, and personalized visual prompts to enhance recognition and grounding.

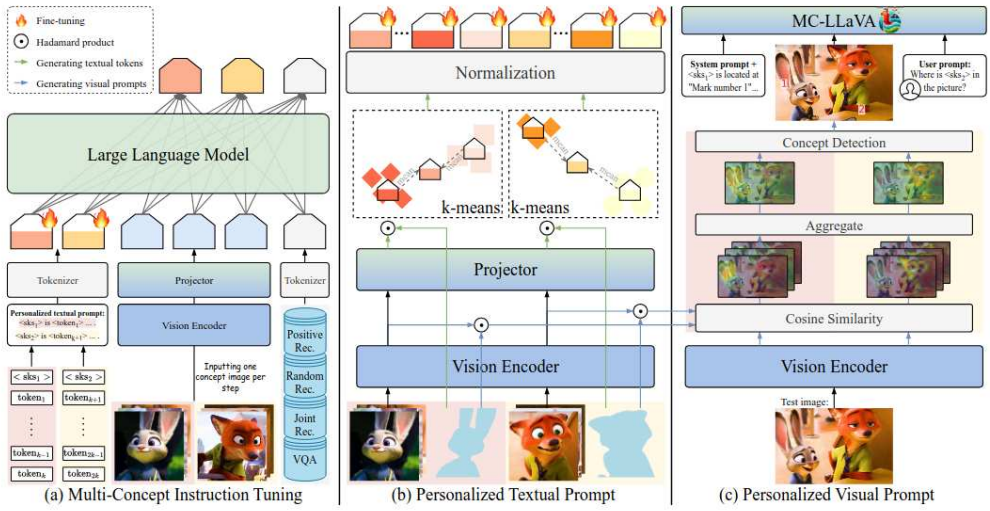


Figure 2.3: MC-LLaVA overview.(a) Multi-concept joint training strategy is used to learn the personalized textual prompts and classifier weights. (b) All concept images are passed through the VLM vision encoder and projection layer, and the projected vision embeddings is used to initialize the m × (k + 1) concept tokens in personalized textual prompts, reducing the costs associated with joint training. (c) During inference, a personalized visual prompt for VLMs is introduced by aggregating location confidence maps based on learned concept tokens. *Figure adapted from An et al. (2024).*

Multi-concept instruction tuning. Given a pre-trained vision-language model and multiple user-provided concepts, the objective of this step is to introduce new concepts by expanding the model's vocabulary and learning personalized textual prompts, while preserving the original model's knowledge. The proposed joint training strategy simultaneously learns personalized textual prompts and classifier weights for multiple concepts.

For m concepts $\{C_j\}_{j=1}^{m}$, each associated with n images $\{I_i\}_{i=1}^{n}$, we define $k+1$ learnable tokens per concept as:

$$\bigcup_{j=1}^{m} \{\langle \texttt{sks}_j \rangle, \langle \texttt{token}_{j,1} \rangle, \ldots, \langle \texttt{token}_{j,k+1} \rangle\},$$

where each $\texttt{sks}_j$ is a special concept identifier while the latent tokens encode the concept semantics. The vocabulary of the language model is expanded by adjusting the classifier weights W from size $D \times N$ to $D \times (N+m)$, where D is the feature dimension and N is the original vocabulary size.

Following Yo'LLaVA, training samples are formatted as triplets (I, X_q, X_a), where I is the input image, X_q is a question, and X_a is the corresponding answer. MC-LLaVA adopts some of the dialogue patterns from Yo'LLaVA, in particular positive recognition, random recognition, and text only conversation training patterns. Additionally it introduces a novel joint recognition pattern. For each current concept $\texttt{sks}_{\text{curr}}$, the model queries whether other concepts $\texttt{sks} \neq \texttt{sks}_{\text{curr}}$ are present in the training images of the current subject, thereby generating joint negative question-answer pairs. This inter-concept negative sampling strategy produces at least $m \times (m-1) \times n$ negative samples in a scenario with m concepts and n images each.

To better understand the implications of this sampling strategy, consider a concrete example with $m = 3$ concepts, denoted as C_1, C_2, C_3, each associated with n positive training images. Starting with concept C_1, for each of its n positive images, the model generates negative joint recognition queries by asking whether the other concepts C_2 and C_3 are present. Since these concepts are not actually present in the images of C_1, the resulting question-answer pairs are valid negative samples. This procedure is repeated for each concept in the set. As a result, for each of the m concepts, $n \times (m-1)$ negative samples are generated, leading to a total of $m \times (m-1) \times n$ joint negative question-answer pairs.

Given these training patterns, the model can be jointly trained on mm subjects simultaneously. The

parameters updated during training in the multi-concept joint training framework are:

$$\theta = \{\langle \texttt{sks}_{1:m} \rangle, \langle \texttt{token}_{1:mk} \rangle, W(:, N+1:N+m)\}.$$

The standard masked language modeling loss is used to compute the probability of the target response X_a:

$$\mathcal{L}(X_a \mid I, X_q, \theta) = -\sum_{t=1}^{T} \log P(X_{a,t} \mid I, X_q, X_{a<t}, \theta),$$

where T is the length of the answer, $X_{a,t}$ is the t-th word in the answer, and $P(X_{a,t} \mid I, X_q, X_a^{<t}, \theta)$ is the conditional probability of predicting the t-th word given the input image, question, previous answer tokens, and parameters θ.

Personalized textual prompts. To reduce the computational cost of joint training and to minimize reliance on high-quality negative samples, MC-LLaVA introduces a novel initialization strategy that leverages visual information. Specifically, all concept images are first passed through the vision encoder of the VLM and a projection layer, and the resulting visual embeddings are used to initialize the concept tokens in the personalized textual prompts.
In the following, we provide a detailed breakdown of the initialization procedure. Given a set of training images $\{I_i\}_{i=1}^{n}$ for each concept, the LLaVA vision encoder $\mathcal{E}_{\text{CLIP}}(\cdot)$ and the projection module P_{MM} are used to obtain projected visual tokens $\{F_i^{\text{MM}}\}_{i=1}^{nhw}$.
To eliminate background noise, Grounded-SAM [21] is applied with the prompt *'the main character in the image'* to generate masks for every training image. The concept-relevant tokens $\{\tilde{F}_i^{\text{MM}}\}_{i=1}^{l}$ are extracted via element-wise Hadamard product between the projected embeddings and the corresponding masks.
Finally, k-means clustering is applied to the compressed visual tokens to obtain k cluster centers $\{K_i\}_{i=1}^{k}$. The special token $\langle \texttt{sks} \rangle$ is computed as the mean of these cluster centers. Consequently, the final concept token set for each concept has shape $(k+1) \times D$.

Personalized visual prompts. The Personalized Visual Prompt is a training-free, inference-time method introduced to enhance concept localization in multi-concept scenarios. It builds upon the *Set-of-Mark (SOM)* [22] framework and consists of two primary stages: generating a location confidence map and constructing the visual prompt.
In a multi-concept setting with m concepts, during training, for each concept C_j, a set of filtered features $\{\tilde{F}_i^{\text{CLIP}}\}_{i=1}^{l_{C_j}}$ is stored. These features are obtained by applying the LLaVA vision encoder $\mathcal{E}_{\text{CLIP}}(\cdot)$ followed by a masking operation to the concept-specific training images.
Given a test image I_t, we extract its encoded features $F_t \in \mathbb{R}^{hw \times c}$ using $\mathcal{E}_{\text{CLIP}}(\cdot)$. For each concept C_j, we compute the cosine similarity between the test image feature F_t and the stored concept features:

$$S_{C_j}^{(i)} = \frac{F_t \cdot \left(\tilde{F}_i^{\text{CLIP}}\right)^{\top}}{\|F_t\|_2 \cdot \|\tilde{F}_i^{\text{CLIP}}\|_2}, \quad i = 1, \ldots, l_{C_j},$$

where $S_{C_j}^{(i)}$ represents the similarity map between the test image and the i-th stored feature of concept C_j. This yields a set of similarity maps $\{S_{C_j}^{(i)}\}_{i=1}^{l_{C_j}}$, each reflecting the spatial distribution of C_j across the test image.
To construct a robust location confidence map, these similarity maps are aggregated via average pooling and normalization:

$$\tilde{S}_{C_j} = \frac{1}{l_{C_j}} \sum_{i=1}^{l_{C_j}} S_{C_j}^{(i)} - \frac{1}{|C|} \sum_{j=1}^{|C|} \left(\frac{1}{l_{C_j}} \sum_{i=1}^{l_{C_j}} S_{C_j}^{(i)} \right),$$

where $|C|$ denotes the total number of concepts. The first term computes the average similarity map for concept C_j, while the second term subtracts the global average across all concepts, removing bias

from varying activation magnitudes and emphasizing relative rather than absolute similarity.
To verify the presence and location of concept C_j in the test image, we evaluate whether a sufficient proportion of pixels in $\tilde{S}_{C_j}$ exceed a confidence threshold τ. If this proportion exceeds a predefined minimum presence ratio γ, we consider C_j present in the image. The pixel with the highest confidence score in $\tilde{S}_{C_j}$ is selected as the representative location for C_j. If the condition is not met, no visual prompt is generated for that concept.
Finally, for the detected $\tilde{m}$ concepts $\{C_j\}_{j=1}^{\tilde{m}}$ in the test image, to create the *SOM* we generate a system prompt in the following format:

$$\texttt{<sks}_j\texttt{> is located at ``Mark Number } j\texttt{''.}$$

This visual prompt effectively conveys concept localization to the model, thereby improving both recognition and grounding performance in multi-concept scenarios.

2.2 Training-Free Personalization Approaches

2.2.1 Retrieval-Augmented Personalization

The Retrieval-Augmented Personalization (RAP) framework [12] introduces a flexible, training-free method for personalizing multimodal large language models (MLLMs). Starting from a general-purpose MLLM, RAP enables its transformation into a personalized assistant without training any tokens, while remaining competitive with training-based methods such as Yo'LLaVA [8] and MyVLM [7].
While those approaches often require numerous labeled images and a large pool of negative samples, resulting in high data collection and computational overhead, RAP achieves personalization using only one positive example images per concept, with no additional training needed.

RAP: Method

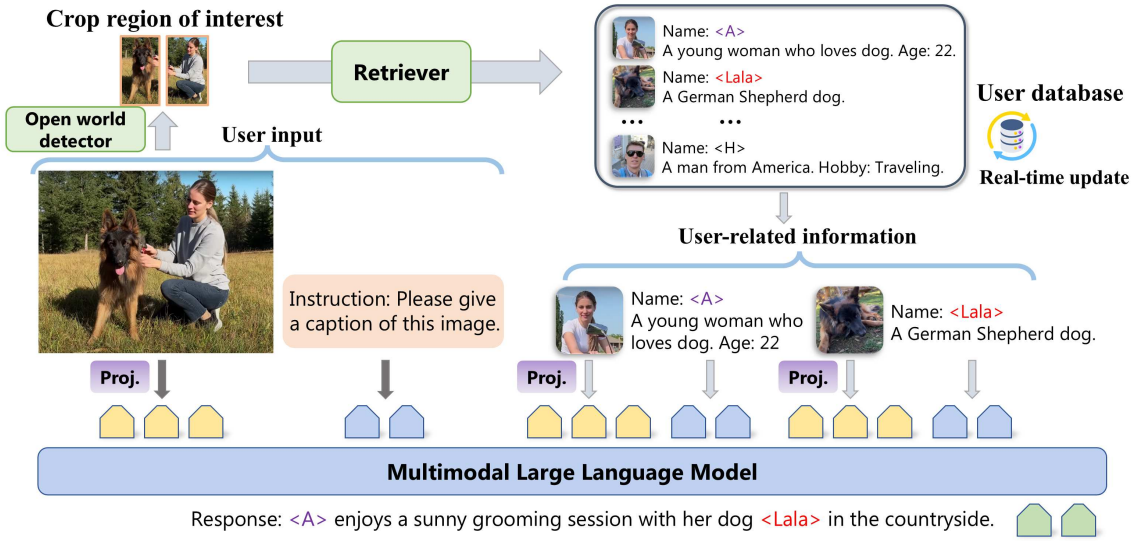


Figure 2.4: Retrieval-Augmented Personalization Framework. Regions of interest, detected by an open-world object detector, are used to retrieve relevant concepts from the memory database. The corresponding images and textual descriptions of the retrieved concepts are then integrated into the input of the multimodal large language model (MLLM) to guide personalized generation. *Figure adapted from Hao et al. (2024).*

This section presents the implementation of the RAP framework. Unlike previous approaches that rely on extensive data collection and additional training to incorporate new concepts, RAP allows pre-trained MLLMs to adapt to diverse users and novel subjects without any further fine-tuning.

The RAP framework is structured into three sequential stages: Remember, Retrieve, and Generate. An overview of this process is illustrated in Figure 2.4.

In the Remember stage, a memory database $\mathcal{M}$ is built to store personalized concepts. Each entry of the database consists of an image I_j, a corresponding name, and a brief textual description T_j of the personalized concept. To enable retrieval, a key k_i is computed for each entry of the database by extracting visual features from the associated image I_j using a pre-trained image encoder $\mathcal{E}(\cdot)$. These keys serve as references for identifying relevant concepts during inference.

When a user initiates a conversation, the input can be denoted as $Q = (X_v, X_q)$, where X_v is the input image and X_q is a textual instruction. For example, X_v might show a girl with her dog, while X_q could request a caption describing the image.

The first stage in the RAP pipeline involves identifying whether any personalized concept stored in the memory database is present in the input image. To enable robust generalization across diverse visual inputs, RAP employs a universal object detection model YOLO [23] as the recognition module $\mathcal{R}(\cdot)$. This module detects potential regions of interest that may correspond to known personalized concepts. The predefined setting P lists the categories to be recognized and remembered, based on the concepts stored in the database. For instance, if the database contains photos of a specific dog, a particular girl, and later also her cat, P would include these personalized categories to guide recognition and memory throughout the task. The setting P conditions the recognition model, which then extracts the relevant region of interest from the input image: $X_u = \mathcal{R}(X_v, X_q \mid P)$. For instance, if the database includes the category *dogs*, the recognition model $\mathcal{R}$ will specifically focus on detecting dogs within the input image.

In the Retrieve step, the identified region of interest is used as a query to retrieve relevant concepts from the database. For each recognized component X_u^i, the corresponding image crop is passed through the image encoder $\mathcal{E}(\cdot)$ to obtain its visual feature vector $v_i = E(X_u^i)$. Then, the Euclidean distance between v_i and each key $k_j \in M$ in the memory database is computed as $\text{Dist}(v_i, k_j) = \|v_i - k_j\|$. The Top-$K$ image-text pairs from the database $\{(I_1, T_1), (I_2, T_2), \ldots, (I_K, T_K)\}$ with the smallest distances are selected as the most relevant matches.

For example, if the database contains multiple dog images and the query image includes a dog, the algorithm retrieves all dog entries from the database and ranks them by similarity to the detected dog region, selecting the Top-K closest matches.

In the Generate stage, each of the Top-K retrieved image-text pair $M_j = (I_j, T_j)$ provides personalized information about the user's concept, which is incorporated into the input of the multimodal large language model. For example, in LLaVA, the retrieved image I_j is first encoded by a pretrained CLIP vision encoder, producing visual tokens Z_j. These tokens are then projected through a learned layer into language-compatible tokens H_j^v, enabling the language model to interpret the visual information. At the same time, the associated textual description T_j is converted into text tokens H_j^q. The user's input image X_v and textual query X_q undergo the same encoding process. All resulting tokens, both visual and textual, are combined and fed into the MLLM, which generates a personalized language response.

During training, the detector and retriever parameters remain frozen, while only the MLLM parameters θ are optimized. Given an output sequence length L, the probability of generating the target answer X_a is computed as:

$$p_\theta(X_a) = \prod_{i=1}^{L} p_\theta(X_{a,i} \mid X_v, X_q, M_1, \ldots, M_K, X_{a,<i})$$

where the model conditions each token on the input, the retrieved concepts, and previously generated tokens.

It is worth noting that, as the number of learned concepts increases, recognition errors also tend to increase, which can lead to a decline in overall performance. Nevertheless, RAP-MLLMs consistently achieve the highest performance across various settings compared to other methods such as MyVLM [7] and LLaVA [16], demonstrating robustness even as the complexity of personalization grows.

The RAP framework presents notable strengths and limitations. On the positive side, by incorporating external personalized information through retrieval, RAP enables MLLMs to generate more specific

and user-tailored responses without requiring additional training. This makes RAP a practical and flexible approach for real-world applications, allowing for real-time updates to the user database, which improves adaptability and responsiveness.
However, this approach also introduces challenges. The increased context length resulting from integrating retrieved data imposes higher computational costs during inference, potentially affecting scalability and latency. Moreover, the overall personalization quality heavily relies on the retriever's accuracy in selecting relevant concepts. Any retrieval errors or mismatches can degrade the relevance and fidelity of the generated responses, limiting RAP's effectiveness.

Chapter 3

Text to image personalized generation

3.1 Introduction to text to image personalized generation

This chapter presents the development of MP-Bench, a dataset designed to facilitate the study of multi-subject personalization. The construction of a dedicated dataset for multi-subject personalization is motivated not only by the limited availability of publicly accessible dataset supporting this setting, but also by the inherent challenges associated with collecting real-world data for this task. These challenges include the requirement for a large number of diverse examples per subject, coverage across a wide range of poses and contexts, and the complexity introduced by the presence of multiple subjects within a single scene. Constructing a dataset of this nature would require gathering a large number of high-quality images for each individual subject, capturing them in a wide variety of poses, lighting conditions, and contexts. These requirements quickly become impractical when scaling to multiple subjects. Moreover, real-world data collection introduces significant ethical, legal, and privacy concerns, especially when the subjects are people.
In contrast, synthetic data generation offers a controlled and scalable alternative. It allows for the creation of diverse and realistic compositions of multiple subjects under varying scenarios, without the need for subject-specific data collection or annotation.
For these reasons, we opted to construct a dataset using personalized text-to-image (PT2I) generation techniques, specifically diffusion-based models, as a means to simulate and control the visual appearance of multiple distinct identities within the same image. The focus in this chapter is specifically on evaluating and exploiting generative models in multi-subject settings, where multiple distinct identities, such as individuals, animals, or objects, must be preserved and faithfully composed within a single image. For instance, given a few reference images of two individuals, A and B, the model should be able to generate realistic representations of both individuals appearing together within the same image, across a variety of settings and environments. Ideally, the model should be capable of scaling to generate images featuring four or five individuals simultaneously within the same scene.
The dataset construction process was guided by the technical capabilities of the MS-Diffusion method [11]. MS-Diffusion was selected for its support of multi-image personalized generation, its ability to incorporate layout guidance, and its operation in a zero-shot setting. This eliminates the need for subject-specific fine-tuning, thereby significantly simplifying and accelerating the dataset generation process.

The remainder of this section is organized as follows: it begins with a general overview of text-to-image generation, with particular emphasis on recent advances enabled by diffusion models. The focus then shifts to the task of personalized generation, outlining its specific challenges and surveying recent methodological progress. This is followed by a detailed presentation of the MS-Diffusion model. The section concludes with a description of the dataset's design, emphasizing its role in the benchmarking of personalized text-to-image systems across both single- and multi-concept scenarios.

3.2 Text-to-image generation

Text-to-image (T2I) generative models are capable of producing high-quality images from user-provided text prompts. With the advancement of deep learning, the T2I task has become one of the most striking applications in the field of computer vision.

T2I generation has its origins around 2015 and has evolved significantly over time (Figure 3.1), driven primarily by three major methodological paradigms: GAN-based models, autoregressive models, and, more recently, diffusion-based models. While all three approaches have contributed to progress in the field, diffusion-based models have gained increasing prominence in recent years, achieving unprecedented success in generating high-quality images.

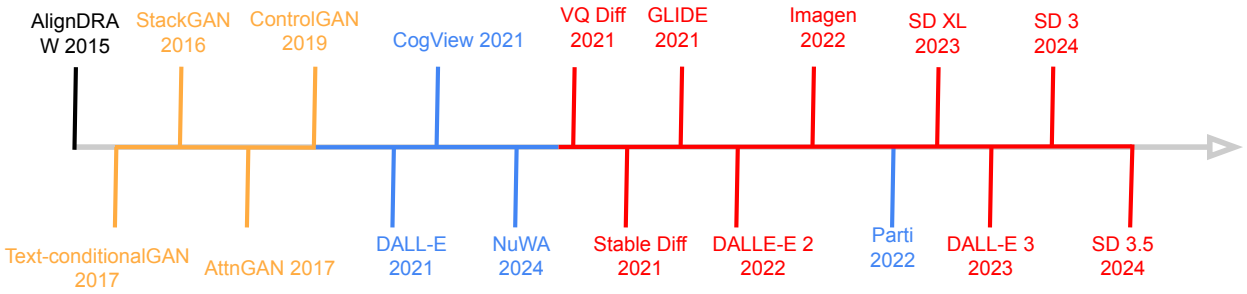


Figure 3.1: Representative works in the text-to-image generation task over time. GAN-based methods, autoregressive methods, and diffusion-based methods are highlighted in yellow, blue, and red, respectively. For brevity, Stable Diffusion is abbreviated as SD in the figure.

Text-conditional GANs [24] were among the first fully end-to-end differentiable architectures, enabling generation from character-level input to pixel-level output. However, these models were typically trained on small-scale datasets, limiting their generalization capabilities. This is mainly due to the instability of GAN training, which hinders the ability to train such models on large-scale datasets.

Autoregressive approaches later leveraged large-scale training data for T2I generation, as exemplified by OpenAI's DALL-E [25]. Despite their success, these methods are inherently limited by high computational costs and the accumulation of sequential errors [25], [26], [27], [28].

More recently, diffusion models have emerged as the state-of-the-art in text-to-image generation, offering significant improvements in fidelity, diversity, and scalability. A milestone in this domain is Stable Diffusion, a prominent framework built upon the Latent Diffusion Model architecture [29]. Stable Diffusion employs a two-stage process: in the first stage, it uses a VQ-GAN encoder to map images into a discrete latent space. Compared to earlier VQ-VAE approaches such as those adopted in DALL-E 2 [30], VQ-GAN integrates an adversarial training objective, which enhances the visual realism of reconstructions. In the second stage, the model performs denoising in latent space to generate coherent image representations from text prompts. This latent-space-based diffusion has been shown to outperform pixel-space alternatives in terms of both efficiency and output quality.
A key component in Stable Diffusion is its use of CLIP embeddings [17] for conditioning. CLIP enables the model to handle a broad variety of textual inputs by mapping them to semantically meaningful latent vectors. The integration of cross-attention layers allows the model to align image synthesis closely with textual descriptions.
In addition to the Stable Diffusion family, which includes versions 1.4, 1.5, 2.0, 2.1, XL, and 3, other representative latent diffusion models include DALL-E 2 [30] and DALL-E 3, which also leverage discrete latent representations and transformer-based decoding strategies for high-quality generation.
While powerful at generating a wide range of diverse concepts and visual styles, general-purpose T2I

models still face challenges when asked to represent specific concepts of interest, such as personalized identities or user-specific objects. These limitations motivate the investigation of personalization techniques, which aim to specialize generation capabilities without retraining the entire model from scratch.

3.3 Diffusion based personalized text-to-image generation

Personalized text-to-image generation focuses on creating images that incorporate the user-specified concept (e.g., specific subjects, faces, or pets) and adhere to provided context. The term *"personalization"* in personalized text-to-image generation can refer to various aspects, such as subject-driven personalization (aimed at generating images that depict a specific subject based on reference images), face-driven personalization (focused on generating human-centric or identity-preserving images), or style-driven personalization (targeting the replication of visual styles from reference images). In following sections, we focus specifically on subject-driven PT2I.

Among the works focusing on subject-driven personalized text-to-image generation, several are particularly noteworthy. Textual Inversion [31] learns a novel word embedding to represent the target subject. Once optimized, this embedding can be used in a plug-and-play fashion for subject-specific generation, without altering the underlying priors of the T2I model. DreamBooth [32] introduces a rare token as a unique identifier for the subject and employs prior-preserving fine-tuning to align this identifier with the reference images. IP-Adapter [33] proposes a zero-shot personalization framework by projecting image embeddings directly into the cross-attention layers of the diffusion model, enabling subject conditioning without additional fine-tuning. Parameter tuning methods such as DreamBooth demonstrate improved representational capacity for capturing fine-grained subject details, however they often come with increased computational costs and potential degradation of pretrained priors, particularly when trained on limited datasets. To mitigate these issues, Custom Diffusion [34] selectively identifies critical parameters for personalization by analyzing a collection of pretrained DreamBooth models. Orthogonal fine-tuning strategies [35], [36] have also been proposed to enhance generalization while minimizing interference with existing model capabilities. In addition to single-subject approaches, a line of research has specifically addressed multi-concept personalization, which aims to generate images that incorporate multiple user-specified subjects, e.g. *"a dog* and a cat*"*, where each concept (dog*, cat*) is defined by reference images. Notable examples include SVDiff [37], which introduces a simple yet effective Cut-Mix-Unmix technique to synthesize training images containing multiple objects, and Custom Diffusion [34], which proposes merging fine-tuned key and value matrices through constrained optimization to jointly represent multiple concepts. To mitigate spatial conflicts between subjects, some approaches incorporate additional layout conditioning (e.g., bounding boxes or keypoints), as exemplified by MS-Diffusion [11], which leverages layout guidance to preserve spatial consistency in multi-subject generation.

3.4 MS-Diffusion model

Personalized text-to-image models have achieved remarkable progress; however, generating images containing multiple personalized subjects remains a significant challenge. Existing methods frequently encounter issues such as subject conflicts, loss of fine details, and inadequate control over image layout. To address these limitations, MS-Diffusion [11] proposes a novel zero-shot framework designed to support multi-subject image generation with explicit layout guidance. Instead of relying on computationally expensive fine-tuning, MS-Diffusion incorporates a Grounding Resampler and a Multi-Subject Cross-Attention mechanism, which together enable precise spatial anchoring and effective disentanglement of individual subject representations.

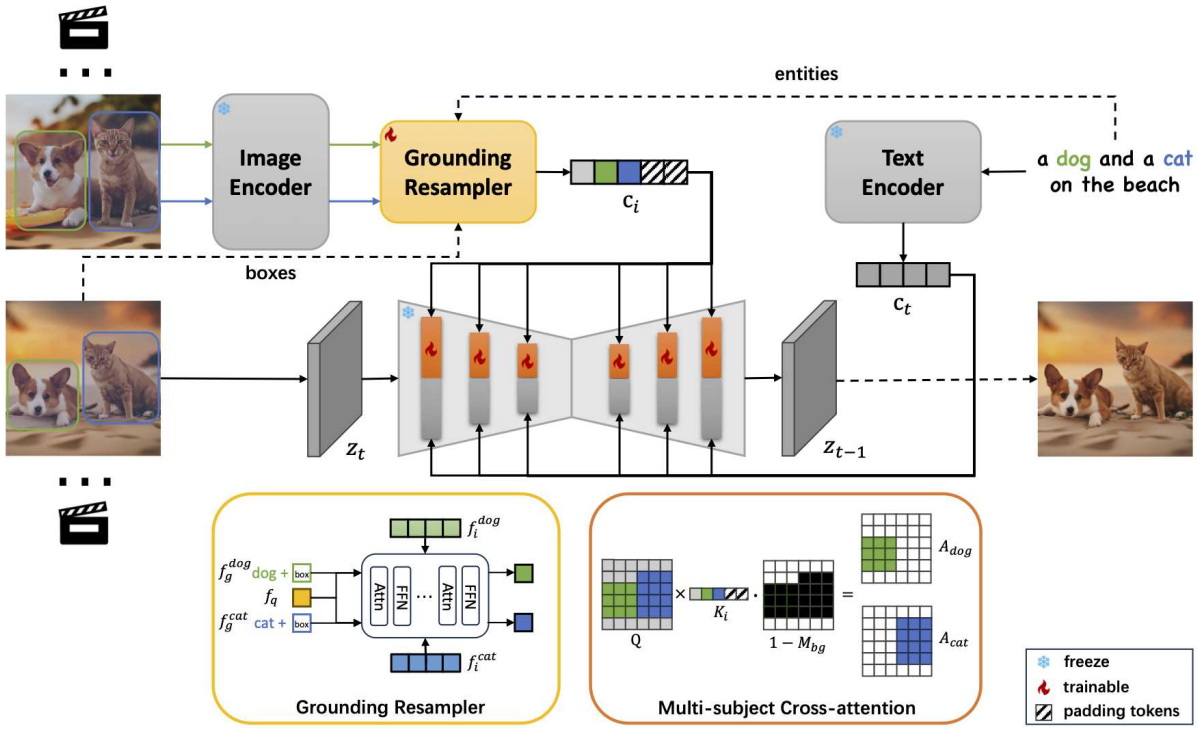


Figure 3.2: **Overview of the MS-Diffusion architecture.** MS-Diffusion model builds upon a Stable Diffusion XL backbone and introduces a dual-conditioning mechanism based on both textual (c_t) and visual-semantic (c_i) inputs. A novel Grounding Resampler module is employed to construct rich image-condition embeddings by extracting detailed subject-specific features from image patches (f_i), and fusing them with grounding information composed of phrase embeddings and Fourier-encoded bounding boxes. This integration enriches the extracted representations by injecting explicit semantic and spatial priors. The resulting latent queries c_i serve as visual condition tokens. Additionally, a multi-subject cross-attention mechanism is introduced in the UNet's transformer layers, enabling fine-grained interactions between the diffusion latent z_t and the grounded visual tokens. *Figure adapted from Wang et al. (2024).*

3.4.1 Introduction to Stable Diffusion with image prompt

MS-Diffusion is built upon the Stable Diffusion framework [29], a powerful latent text-to-image generative model that differs fundamentally from earlier diffusion models operating in pixel space. By performing the denoising process in a compressed latent space, this model achieves faster and more efficient generation while maintaining high image fidelity.
Stable Diffusion architecture, depicted in figure 3.3, comprises three main components: a Variational Autoencoder (VAE); an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$, which transform images between pixel and latent space; a denoising U-Net, which performs iterative noise removal within the latent domain; finally a conditioning module.
The image generation process (as depicted in figure 3.2) begins with a Variational Autoencoder that

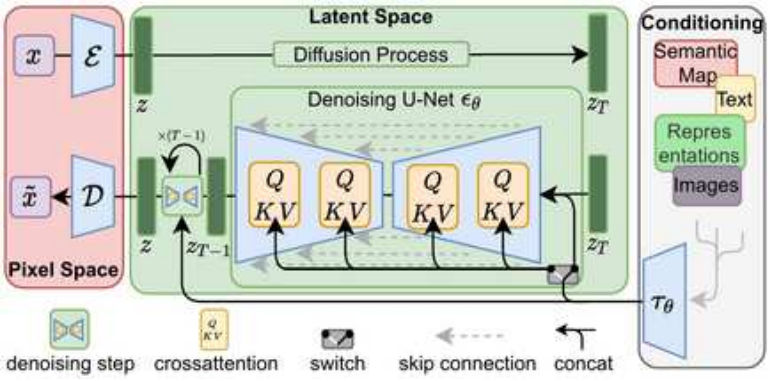


Figure 3.3: **Stable Diffusion architecture overview**. The model operates in latent space, where an input image is first encoded by a Variational Autoencoder into a compact latent representation. During the forward diffusion process, Gaussian noise is progressively added to the latent. A denoising U-Net, conditioned on a text prompt y via a cross-attention mechanism, learns to iteratively remove the noise. After completing the reverse diffusion steps, the denoised latent is decoded back into pixel space by the VAE decoder to produce the final image. *Figure adapted from Rombach et al. (2021).*

encodes an input image $x \in \mathbb{R}^{H\times W\times 3}$ into a lower-dimensional latent representation $z = \mathcal{E}(x)$. This latent captures the essential structure and semantics of the image, which can then be used as the basis for generation. During training, Stable Diffusion learns through a denoising diffusion probabilistic process composed of two phases: the forward and reverse diffusion. In the forward phase, Gaussian noise is incrementally added to the latent representation over multiple time steps $t = 0, .., T$, progressively degrading the image. This process produces a series of latent variables $z_1, z_2, \ldots, z_T$, where:

$$z_t = \sqrt{\alpha_t} z_0 + \sqrt{1-\alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I).$$

Here, α_t are predefined noise schedules controlling the amount of noise at step t.
The reverse phase, occurring within the U-Net module and constitutes the core of the model, is where generation occurs. During this phase the model progressively removes noise from the corrupted latent representation by predicting the noise component at each step.
The U-Net outputs a noise estimate, which is scaled and subtracted from the current latent z_t to produce a cleaner latent z_{t-1}. After T such steps, the final latent is decoded by the VAE decoder $\mathcal{D}$ to reconstruct the denoised image $\hat{x} = \mathcal{D}(z_0)$.
During training, the model learns to invert the noising process by minimizing the discrepancy between the predicted noise $\hat{\epsilon}_\theta$ and the actual noise ϵ added in the forward diffusion. This is formalized through the objective:

$$\mathcal{L}_{\text{LDM}} := \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),c,t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c)\|_2^2\right]$$

where c represent the additional condition, such as text or an image.
Once trained, Stable Diffusion can generate novel images by sampling random noise $z_T \sim \mathcal{N}(0, I)$ and running the learned reverse diffusion process, optionally conditioned on a text prompt or other

modalities.
The U-Net module is based on a ResNet backbone and consists of three key components per layer: a residual block, which processes spatial features, a self-attention block, which allows long-range dependencies between latent features, and a cross-attention block, which injects conditioning information—typically from a text prompt. In our scenario the U-Net architecture, is conditioned on both image and text prompts via a cross-attention mechanism. This mechanism enables the model to associate specific image regions with corresponding elements of the prompt, ensuring alignment between text and visual content. The text prompts are first encoded using a pretrained CLIP text encoder (specifically, ViT-L/14), which transforms them into high-dimensional semantic embeddings. These embeddings are injected into multiple layers of the U-Net, guiding the generative process so that the final image faithfully reflects the input description. Given query features Z and text features c_t, the output of the cross-attention operation, Z', is computed as:

$$Z' = \text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V$$

where $Q = ZW_q$, $K = c_tW_k$, and $V = c_tW_v$ represent the query, key, and value matrices respectively, obtained via trainable linear projections with weight matrices W_q, W_k, and W_v. To incorporate based image conditioning, an additional cross-attention layer is introduced in parallel to each of the original cross-attention layers in the UNet model. This extension allows the model to jointly attend to visual concepts specified by image features c_i. The image cross-attention output Z'' is defined analogously as:

$$Z'' = \text{Attention}(Q, K', V') = \text{Softmax}\left(\frac{QK'^\top}{\sqrt{d}}\right)V'$$

Here, $Q = ZW_q$, $K' = c_iW'_k$, and $V' = c_iW'_v$ are the query, key, and value matrices, respectively, with W'_k and W'_v representing the learnable weight matrices for projecting the image features.

At this stage, both the image features c_i and text features c_t are integrated into the UNet backbone through the joint cross-attention mechanism. This mechanism is designed to effectively combine the contributions of both conditioning modalities, visual and textual, by computing a weighted fusion of their respective attention outputs. The final formulation is as follows:

$$\text{Attn}(Q, K_i, K_t, V_i, V_t) = \gamma \cdot z_{\text{img}} + z_{\text{txt}} = \gamma \cdot \text{Softmax}\left(\frac{QK_i^\top}{\sqrt{d}}\right)V_i + \text{Softmax}\left(\frac{QK_t^\top}{\sqrt{d}}\right)V_t$$

In this formulation, $Q = zW_q$ is the query matrix derived from the UNet's intermediate feature representation z, while $K_i = c_iW_k^i$, $V_i = c_iW_v^i$, $K_t = c_tW_k^t$, and $V_t = c_tW_v^t$ are the key and value matrices for image and text features, respectively. W_q, W_k, and W_v denote the corresponding projection weight matrices, and d is the dimensionality of the key vectors. The scalar γ is a learned or fixed weighting factor used to modulate the influence of the image-based attention output z_{img} relative to the text-based output z_{txt}.

3.4.2 Grounding resampler

In MS-Diffusion, a Grounding Resampler is proposed as an alternative and improved strategy for projecting image features. While text embeddings are typically compact and dense, image embeddings derived from visual encoders are high-dimensional, spatially distributed, and rich in detail [11]. This complexity makes the projection into the conditioning space challenging, leading to the loss of important subject information. To address this, the Grounding Resampler is designed to selectively extract subject-relevant visual features and integrate them with structured grounding signals, such as semantic labels and bounding boxes. Rather than relying on standard pooling or projection layers, it employs a set of learnable queries that attend to the image features, capturing only the components needed to condition the generation process.

In multi-subject scenarios, where each subject is defined by both its identity and spatial location, the Grounding Resampler plays a crucial role by enabling precise control through the alignment of each subject's representation with its corresponding region and semantic label.
The Grounding Resampler operates as a cross-attentional module that transforms high-dimensional image features into a compact set of latent representations, guided by semantic and spatial grounding cues. The input to the module consists of a sequence of image embedding $f_i \in \mathbb{R}^{N \times D}$. To extract task-relevant visual features, the resampler introduces a set of learnable query tokens $f_q \in \mathbb{R}^{M \times D}$, which are either randomly initialized and learned end-to-end (in the *random* mode), or computed dynamically from grounding information (in the *grounding* mode). In the latter case, each latent token is derived by concatenating a phrase embedding $f_p \in \mathbb{R}^{M \times D_p}$, encoding the semantic information of the subject, with a Fourier-encoded bounding box embedding $f_b \in \mathbb{R}^{M \times D_b}$, which indicate the areas where the subjects are supposed to be. The resulting concatenated embedding is passed through a multi-layer perceptron to match the internal hidden dimension. This MLP is fully learnable and is defined as a composition of two linear layers with a GELU activation in between. Once both f_i and f_q are available, the module applies a stack of L attention layers, each consisting of a cross-attention block followed by a feed-forward network. At each layer ℓ, the attention operation is computed as:

$$\text{RSAttn} = \text{Softmax}\left(\frac{\text{Q}\left(f_q\right)\text{K}^\top\left([f_i, f_q]\right)}{\sqrt{d}}\right)\text{V}\left([f_i, f_q]\right)$$

where RSAttn refers to the grounding resampler attention while Q, K, and V are the learnable projection matrices for queries, keys, and values, respectively, and are shared across heads in the multi-head attention formulation. Finally, after all L layers, the output latent tokens are normalized and passed through an output projection. For an input containing n distinct subjects, the Grounding Resampler processes each subject independently with respect to its semantic and spatial grounding information. That is, the projection pipeline—including the generation of query tokens from phrase embeddings and Fourier-encoded bounding boxes—is applied separately for each subject without any cross-interaction. This subject-wise independence ensures that the representation of each entity remains disentangled and specific to its grounding inputs. The output of this process is a set of n latent query groups, each consisting of n_t tokens. These are concatenated along the sequence dimension to form a single composite tensor $c_i \in \mathbb{R}^{(n \cdot n_t) \times D}$, which is subsequently passed to the diffusion model as the conditioning signal. This structure enables the model to condition generation on multiple grounded subjects simultaneously.

3.4.3 Multi-Subject Cross-attention

In multi-subject generation scenarios, a key challenge is managing semantic conflicts and spatial entanglements between distinct subjects and the background. Traditional diffusion models often struggle to maintain subject integrity when multiple entities co-exist in the same scene, leading to artifacts, blending, or misalignment between the subjects and their intended spatial locations.
To address this, MS-Diffusion introduces a Multi-Subject Cross-Attention mechanism. Implemented as a modification of the cross-attention layers in the U-Net's denoising network, the Multi-Subject Cross-Attention mechanism extends the standard diffusion attention by integrating spatially-aware attention masks. These masks, denoted as M_j for each subject j, act as spatial filters applied to the attention computation between image queries and condition keys. For each subject j, the binary mask $M_j \in \{0, -\infty\}^{H \times W}$ is constructed by transforming the bounding box onto the latent resolution grid. All positions inside the box are set to 0 (allowing attention), and those outside are set to $-\infty$ (effectively masked out in the softmax). These masks are then flattened and concatenated to form the global matrix $M = \text{Concat}(M_0, M_1, \ldots, M_n)$.
Given the matrix M, the conditional image latent $\hat{z}_{\text{img}}$ is computed as:

$$\hat{z}_{\text{img}} = \text{Softmax}\left(\frac{QK_i^\top}{\sqrt{d}} + M\right)V_i$$

where K_i, V_i are the key and value matrices derived from the conditioned image tokens.
To handle background regions (meaning areas that are not covered by any subject's bounding box) the model introduces dummy tokens representing background content. These tokens absorb the attention from overlapping or unassigned queries and help maintain balance between layout-guided image content and free-form textual guidance. A background mask M_{bg} is subsequently applied to suppress these tokens when not required:

$$z_{\text{img}} = (1 - M_{\text{bg}}) \cdot \hat{z}_{\text{img}},$$

where M_{bg} is articulated as a binary mask, with elements within the subject bounding boxes designated as zero. Through this design, MS-Diffusion explicitly allocates spatial attention to individual subjects, promoting clearer boundaries and reducing ambiguity. This structured attention mechanism allows the model to better resolve subject conflicts, improve compositionality, and maintain textual fidelity in complex multi-object scenes.

3.4.4 MS-Diffusion Training Procedure and MS-Bench Dataset Construction

During training, the parameters of the pre-trained SDXL backbone (namely the U-Net, the VAE, and the text and image encoder) are kept frozen, while only the newly introduced multi-subject cross-attention layers and grounding resampler modules are optimized.
The MS-Bench dataset is constructed from video clips by selecting two frames: one as reference and the other as ground truth. Each video is sampled multiple times in different ways, 2 to 5 frames for each subject are selected and used for training.
Captions are generated using BLIP-2, and subject entities are extracted via a Named Entity Recognition model. Bounding boxes are obtained with Grounding DINO and refined into segmentation masks using SAM. Subject correspondence across frames is established through a Hungarian Matching algorithm on visual embeddings. Each sample contains up to four subjects, with filters applied to remove unbalanced cases.
The final dataset comprises approximately 3.6 million video clips, balancing general and product-specific domains, and provides diverse spatial layouts for robust multi-subject generation.
The model is trained using PyTorch 2.0.1 and Diffusers 0.23.1 for $120k$ steps on 16 A100 GPUs, with a batch size of 8 and a learning rate of $1e^{-5}$.

3.5 MP-Bench dataset's design

This section presents the construction details of the multi-subject personalization dataset, refferred to as MP-Bench. As anticipated in the chapter introduction, the motivation for creating this dataset stems from several factors.
The first motivation is the scarcity of publicly available datasets suitable for evaluating multi-subject personalization tasks. In order to assess model performance in this context, a dataset is required that captures individual concepts both in isolation and in combination with other personalized concepts, within a consistent setting. At the time of writing, the only dataset explicitly designed for this purpose is the MC-LLaVA dataset [10].
Second, collecting such a dataset from real-world data quickly becomes combinatorially complex as the number of subjects increases in multi-concept scenarios. This results in extended acquisition times and increasing logistical complexity. Synthetic data, on the other hand, offers scalability, allowing the dataset to be expanded as needed.
Finally, by generating a dataset based on publicly available resources, potential privacy or ethical concerns associated with the acquisition of new real-world data are effectively avoided.
The construction of the personalized dataset began with the selection of a seed dataset, that is, a single-image dataset used as the basis for generating multi-subject images. Although MS-Diffusion is trained on a large-scale video dataset comprising millions of video clips, making it in principle compatible with arbitrary seed image sets, the choice of seed datasets was guided by practical constraints and performance considerations.

Initial experiments with the Yo'LLaVA dataset revealed suboptimal results, principally due to the fact that its target concepts were mostly people, which made it harder for the model to render accurate face shapes and handle hand generation correctly. It was also constrained by limitations related to the pre-training scale of MS-Diffusion, which further reduced its effectiveness as a seed dataset. Consequently, the Yo'LLaVA dataset could be scaled only up to two subjects per image, forming the first split of the generated dataset: Yo'LLaVA single- and two-subject images.
To achieve better subject compositionality, MS-Bench was selected as a complementary source and served as the seed for the second split of the dataset, which comprises both single- and triple-subject images. Unlike Yo'LLaVA, it proved more suitable for subject compositionality and allowed scaling up to three concepts per image, aligning better with the training objectives outlined in the pipeline.
The Yo'LLaVA dataset comprises images of 40 distinct target subjects, which are organized into five broader categories:

- Person: consists of 10 micro-influencers or personal acquaintances.
- Pets: includes 5 personal pets.
- Landmarks: encompasses 5 local landmarks.
- Objects: contains 15 items sourced from Amazon product reviews.
- Fictional Characters: comprises 5 characters, drawn from movies released in 2023 or supporting roles.

The MS-Bench dataset comprises images spanning 40 categories, which can be grouped into six broad classes:

- Living: ten subjects including cats and dogs of various colors and breeds;
- Objects: ten commonly used items ranging from plush toys to shoes;
- Scenes: four distinct natural background scenes, such as mountains and waterfalls;
- Upper-wear: five images of hats;
- Mid-wear: six images of shirts;
- Lower-wear: three images of pants;
- Full-body wear: two images of dresses.

To generate an image, the model requires as input one or more reference photos of the subject, a text prompt, and a class label corresponding to the seed images. Layout guidance can be optionally provided to spatially condition the subject placement within the image. While not strictly necessary in single-subject scenarios, layout guidance proves particularly useful in multi-subject settings, where it helps resolve spatial conflicts by guiding the model in assigning distinct regions to each subject. Specifying layout guidance requires providing the coordinates of a bounding box, defined by four values: $(x, y, width, height)$. These parameters indicate the top-left corner of the box and its dimensions.
To introduce variability in both the scenarios and the composition of subjects within the image, a diverse set of text prompts was defined. For each generation, one prompt was randomly selected from this set.

3.5.1 Single-concept image generation

For single-image generation, five prompts were selected for each concept within the 40 categories, drawn from the broader set of available prompts (as exemplified in Table 3.2). For layout guidance, a fixed bounding box was used: `boxes = [[0.25, 0.25, 0.75, 0.75]]`, which approximately corresponds to the central region of the image.

Category	**Example Prompts**
Living	• Best quality, high-quality portrait of {concept} at the park • Best quality, high-quality close-up of {concept} watching the sunset by the lake • Best quality, high-quality portrait of {concept} sniffing a flower • Best quality, high-quality close-up of {concept} sitting by a fireplace
Object	• Best quality, high-quality portrait of {concept} on a wooden desk • Best quality, high-quality close-up of {concept} with soft morning light • Best quality, high-quality macro photo of {concept} in a library environment • Best quality, high-quality artistic rendering of {concept} in a modern setting
Scene	• Best quality, high-quality close-up of {concept} at sunrise • Best quality, high-quality portrait of {concept} illuminated at night • Best quality, high-quality close-up of {concept} under stormy clouds • Best quality, high-quality portrait of {concept} surrounded by flowers
Mid-wear	• Best quality, high-quality full view of {concept} on a mannequin • Best quality, high-quality photo of a stylish {concept} draped over a chair • Best quality, high-quality photo of a designer {concept} hanging in a boutique • Best quality, high-quality photo of a well-worn {concept} in a laundry basket
Up-wear	• Best quality, high-quality close-up of {concept} on a hanger • Best quality, high-quality close-up of a carefully folded {concept} on a shelf • Best quality, high-quality close-up of a vintage {concept} laid out on a table • Best quality, high-quality close-up of a stylish {concept} placed next to accessories
Down-wear	• Best quality, high-quality close-up of a pair of {concept} neatly arranged on a bed • Best quality, high-quality close-up of a comfortable {concept} draped over a chair • Best quality, high-quality full view of {concept} hanging in a boutique • Best quality, high-quality close-up of a well-worn {concept} folded in a laundry basket

Category	Example Prompts
Whole-wear	• Best quality, high-quality full view of {concept} on a mannequin • Best quality, high-quality photo of {concept} displayed in a fashion store • Best quality, high-quality full view of a stylish {concept} laid out on a bed • Best quality, high-quality photo of a fashionable {concept} draped over a chair

Table 3.2: Representative prompt examples for each subject category used in the dataset generation. The placeholder `{concept}` is replaced by the actual subject during runtime.

3.5.2 Multi-concept image generation

For multi-image generation, a similar approach was adopted as in the single-image setting. However, in this case, predefined combinations of categories were used, each associated with a specific set of bounding boxes for layout guidance (Tables 3.3, 3.4 and 3.5, 3.6, 3.7). This distinction is necessary, as the spatial arrangement of subjects varies depending on the categories involved. For instance, placing two objects in the same image requires different bounding boxes compared to a combination such as upper-wear and lower-wear.
As previously discussed, the Yo'LLaVA dataset supports up to two subjects per image, while the MS-Bench dataset was scaled to accommodate up to three. The following tables present the main category combinations and prompt templates used for generating such multi-subject images.

Category Combination	Prompt Examples
person, person	• Best quality, high-quality portrait of {person1} and {person2} enjoying coffee at a café. • Best quality, high-quality close-up of {person1} and {person2} during a walk. • Best quality, high-quality portrait of {person1} and {person2} relaxing on the sofa. • Best quality, high-quality close-up of {person1} and {person2} listening to music in a park. • Best quality, high-quality close-up of {person1} and {person2} at the cinema. • Best quality, high-quality close-up of {person1} and {person2} cooking together.

Table 3.3: Double-target concept combinations with example prompts. Placeholders `{...}` represent subject insertions. Boxes are omitted from the table and are the same for each combination: (`[0.00, 0.25, 0.50, 0.75]`, `[0.50, 0.25, 1.00, 0.75]`)

Category Combination	**Prompt Examples**
person, pet	• Best quality, high-quality portrait of {person} cuddling {pet} in a cozy living room. • Best quality, high-quality close-up of {person} playing fetch with {pet} on the grass. • Best quality, high-quality portrait of {person} taking {pet} for a walk on the beach. • Best quality, high-quality close-up of {person} feeding {pet} a treat.
pet, pet	• Best quality, high-quality portrait of {pet1} and {pet2} playing in the grass. • Best quality, high-quality close-up of {pet1} and {pet2} napping together. • Best quality, high-quality portrait of {pet1} and {pet2} looking at a butterfly in a garden. • Best quality, high-quality close-up of {pet1} and {pet2} sharing a bowl of food.
cartoon, cartoon	• Best quality, high-quality animated portrait of {character1} and {character2} on a new adventure. • Best quality, high-quality close-up of {character1} and {character2} sharing an ice cream. • Best quality, high-quality portrait of {character1} and {character2} on a road trip. • Best quality, high-quality animated close-up of {character1} and {character2} shopping together.

Table 3.4: Double-target concept combinations with example prompts. Placeholders `{...}` represent subject insertions. Boxes are omitted from the table and are the same for each combination: (`[0.00, 0.25, 0.50, 0.75]`, `[0.50, 0.25, 1.00, 0.75]`)

Category Combination	**Bounding Boxes**	**Examples**

scene, living, object	`[0.00, 0.00, 1.00, 1.00]`
	`[0.25, 0.25, 0.75, 0.75]`
	`[0.65, 0.25, 0.65, 0.75]`
		• Best quality, high-quality {scene} with {living} playing near {object}.
		• {scene} setting where {living} is sleeping near to {object}.
		• {scene} background with {living} resting next to {object}.
		• {scene}, where {living} plays with {object}.

living, object, midwearing	`[0.00, 0.25, 0.50, 0.75]`
	`[0.50, 0.25, 1.00, 0.75]`
	`[0.00, 0.25, 0.50, 0.75]`
		• {living} wearing {midwearing}, posing next to {object}.
		• {living} sniffing {object}, sporting a cozy {midwearing}.
		• {living} tugging at {object}, looking dashing in {midwearing}.
		• {living} in {midwearing}, sitting beside {object}.

scene, object, living	`[0.00, 0.00, 1.00, 1.00]`
	`[0.25, 0.25, 0.75, 0.75]`
	`[0.65, 0.25, 0.65, 0.75]`
		• {scene}, where {object} sits next to {living} enjoying the view.
		• {scene}, {living} curiously approaches {object}.
		• {scene} with {object} beside resting {living}.
		• {scene}, {living} playing near {object}.

Table 3.5: Triplet combinations with example prompts and spatial box definitions for generating dataset. The `{...}` placeholders represent subject insertions.

Category Combination	**Bounding Boxes**	**Examples**

upwearing, wholewearing, downwearing	`[0.25, 0.00, 0.75, 0.25]`
	`[0.25, 0.25, 0.75, 0.60]`
	`[0.25, 0.60, 0.75, 1.00]`
		• A man wearing {upwearing}, {wholewearing}, and {downwearing}.
		• A woman wearing {wholewearing}, with trendy {upwearing} and {downwearing}.
		• A woman wearing {upwearing}, elegant {wholewearing}, and chic {downwearing}.
		• {upwearing}, {wholewearing}, and {downwearing} for a polished look.

object, object, scene	`[0.00, 0.00, 1.00, 1.00]`
	`[0.25, 0.25, 0.75, 0.75]`
	`[0.65, 0.25, 0.65, 0.75]`
		• {object} and {object} in a stunning {scene}.
		• {scene}, {object} rests beside {object}.
		• {object} and {object} in the heart of a {scene}.
		• {object}, {object}, and {scene} in harmony.

living, living, object	`[0.00, 0.25, 0.35, 0.75]`
	`[0.35, 0.25, 0.65, 0.75]`
	`[0.65, 0.25, 1.00, 0.75]`
		• {living}, {living} and {object} are on the table.
		• {object}, {living} rests beside {living}.
		• {living} and {living} looking at {object}.
		• {living} and {living} with {object} beside them in the garden.

Table 3.6: Triplet combinations with example prompts and spatial box definitions for generating dataset. The `{...}` placeholders represent subject insertions.

Category Combination	**Bounding Boxes**	**Examples**

object, object, object	`[0.00, 0.25, 0.35, 0.75]`
	`[0.35, 0.25, 0.65, 0.75]`
	`[0.65, 0.25, 1.00, 0.75]`
		• {object}, {object}, and {object} arranged artistically.
		• A still life of {object}, {object}, and {object}.
		• {object}, {object}, and {object} in perfect symmetry.
		• {object} with {object}, {object} completing the scene.

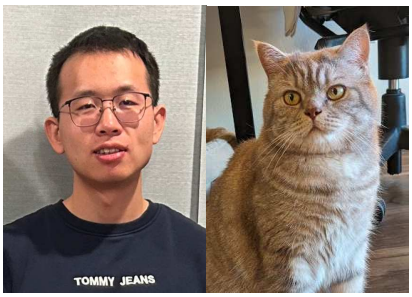
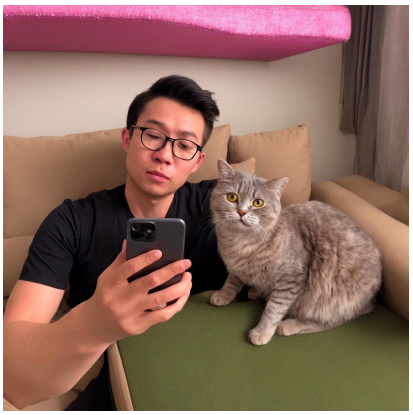
Table 3.7: Triplet combinations with example prompts and spatial box definitions for generating dataset. The `{...}` placeholders represent subject insertions.

3.5.3 Dataset Post-Cleaning

Based on the configurations defined in the previous sections, multiple input settings were generated and used to produce images with the MS-Diffusion model. For the single-target concept setting, each configuration incorporated only one personalized concept at a time. For each concept, MS-Diffusion was prompted to generate personalized images across six different scenarios, each corresponding to a distinct text prompt. Five images were generated per prompt, resulting in 30 initial images per concept.
Similarly, for the multi-concept personalization setting, where m personalized concepts were to be combined within a single image, MS-Diffusion was again prompted with six different text prompts, generating five images per prompt.
A manual post-processing step was subsequently conducted to remove low-quality outputs, based on visual inspection and qualitative assessment. The dataset cleaning phase was guided by a set of qualitative criteria aimed at refining the dataset and ensuring a high standard of visual output. Specifically, the assessment focused on several key aspects: the consistency of the generated subjects with their original personalized representations; the accurate inclusion of the target subjects within the image; the overall visual quality; the degree of realism, particularly favoring outputs resembling photographic imagery; and the level of diversity within each image set, with near-duplicate samples being removed to maintain variability.
In line with the goals of this work, this post-processing step contributed to ensuring a higher overall standard in the final dataset. As anticipated, the number of rejected samples increased with the complexity of the generation task, particularly when combining two or more concepts, reaching a notable peak when four concepts were involved. The final MP-Bench dataset is divided into two splits, corresponding to the two seed datasets employed: Yo'LLaVA and MS-Bench. The Yo'LLaVA-derived split, which supports up to two subjects per image, contains 375 single-subject training images, 333 single-subject test images, and 139 two-subject test images.
The MS-Bench-derived split, which allows for up to three subjects per image, comprises a total of 176 multi-subject test images, 195 single-subject test images, and 273 training single-subject images.

3.5.4 Qualitative results of MS-Diffusion in multi-subject personalization

In the following section, we present a set of qualitative results that highlight the capabilities of MS-Diffusion in the context of multi-subject personalization. This evaluation aims to illustrate how the model effectively integrates multiple personalized subjects into coherent visual outputs, guided by textual prompts.
Each row in the visualization corresponds to a distinct test case. The first column shows the initial seed subjects, which serve as the personalized inputs to the model. The second column contains the guidance prompt provided to MS-Diffusion, which describes the desired composition or semantic content of the generated image. Finally, the third column displays a representative image generated by the model.
In the case of the Yo-llava derived split of the dataset, the bounding boxes used as prompts for the generative model were omitted, as they were identical across all samples and were explicitly indicated in the image captions. Conversely, in the split derived from MS-Bench, the bounding boxes were included, as each sample involved a unique combination of three distinct target concepts from different semantic classes, necessitating the use of a specific bounding box configuration for each generation.
In the following, we first present examples with two personalized subjects per image from the Yo'LLaVA dataset, followed by triplet personalization cases from the MS-Bench.

Yo'LLaVA subjects	Prompt	MS-Diffusion
	portrait of a man cuddling a dog in a cozy living room	
	portrait of a woman cuddling a cat in a cozy living room	
	portrait of a man cuddling a cat in a cozy living room	
	a woman and a man listening to music in a park.	

MS-Diffusion subjects	Prompt	MS-Diffusion
	a cat and a dog rest beside a lantern	
	a dog wearing a coat posing next to a backpack	
	a cat and a cat looking at a lantern	
	a cat and a dog and a flower are on the table	

Subjects	Prompt	MSDiffusion
	a cat and a dog with a duck toy beside them in the garden	
	a cat wearing a jacket posing next to a sloth plushie	
	a cat a dog and a teddybear are on the table.	
	a cat and a cat with a teapot beside them in the garden	

3.5.5 Limitations

The final dataset includes images containing up to three subjects per image. In a limited number of test cases, it was possible to generate images with four subjects, but this proved to be challenging. This limitation appears to originate from intrinsic properties of the MS-Diffusion model as well as the MS-Bench and Yo'LLaVA datasets. Nevertheless, the proposed dataset generation approach retains a key advantage: while the number of subjects per image is constrained, the number of generated samples can be scaled arbitrarily. As a result, the method remains highly scalable in terms of dataset size.
Despite these strengths, the dataset generation process exhibited several failure modes, which can be broadly categorized into the following primary classes of issues:

- Challenges with fine anatomy, especially human hands and animal paws: the generation of detailed appendages such as human hands and animal paws remains a significant difficulty. This problem largely stem from the scarcity of detailed training data of this kind in large-scale datasets such as LAION [19]. In the generated images, hands and paws often appear distorted or unnatural due to the complexity of their articulation and appearance. For example, fingers may be fused together or have incorrect numbers, and paws might lack clear toe separation or appear blurred. Visual examples highlighting these issues are shown in Table 3.4, first row).
- Quality and completeness of seed images: the quality of the seed images used as input to the generative model notably impact the results. For instance, some Yo'LLaVA seed images depict subjects partially cropped or not shown at full height. When MS-Diffusion is prompted to generate images showing the subject's full figure, it often fails to generalize beyond the limited seed input, resulting in incomplete or truncated depictions (see Figure 3.4 at second row for examples).
- Fusion of characteristics among multiple subjects: when generating images with multiple subjects, especially of the same class (e.g., two people), the model occasionally blends distinctive features of different subjects into hybrid or ambiguous representations, rather than preserving clear individual identities. This fusion leads to a loss of distinctiveness even when separate seeds are provided. Please refer to the third row of the Table 3.4 for visual experiments.
- Loss of fine-grained details: certain small but semantically important details, such as text on a cup, logos on hats, or the exact shape of clock hands, are sometimes omitted or altered by the diffusion model. While the overall shape of the object remains recognizable, these subtle features may not be faithfully reproduced, reducing the fidelity of personalized generation. Examples highlighting these issues are shown in Table 3.5, first row)
- Inconsistent real-world size proportions: maintaining accurate size proportions relative to real-world scales presents an additional challenge. Although there is a tunable parameter intended to control subject scale, this adjustment can be unstable and difficult to optimize, leading to inconsistent size relationships among subjects and objects in the generated images.
- Limitations in scaling beyond three personalized subjects: the model exhibits substantial difficulties when generating images with more than three personalized subjects. Successful generation with more than three subjects is only achievable for a limited subset of visually simple or highly distinctive subject types, and even then, only after numerous attempts due to the high rejection rate during the post-cleaning phase. Representative visual examples of these limitations are shown in Table 3.6.

The following section present qualitative examples of failures and issues encountered during the generation process in both single- and multi-concept settings. For each identified challenge, three representative examples are provided.

(a)

(a) Fine anatomy of the human hand. The first row provides examples of common errors in depicting human hand anatomy. In each case, we observe supernumerary fingers, instances of three or more hands, and significant blurring in the hand region. All of these examples are taken from experiments conducted using the Yo'LLaVA dataset.

(b)

(b) Cut figures. The images above show examples of cut figures, that is, humans depicted with the lower half of the body omitted. Interestingly, this occurs more frequently when people are shown in a park or outdoors, particularly on grass.

(c)

(c) Fusion of characteristics among multiple subjects. The above figure shows examples of concept fusion during the generation process. In the first two images from the left, the MS-diffusion model was prompted to generate an image containing two human target concepts. However, it produced three individuals, with the central figure exhibiting characteristics of both targets. In the final image, the model was given images of a personalized cup and a plush toy; the result is a fusion of the cup and the plush toy.

Figure 3.4

(d)

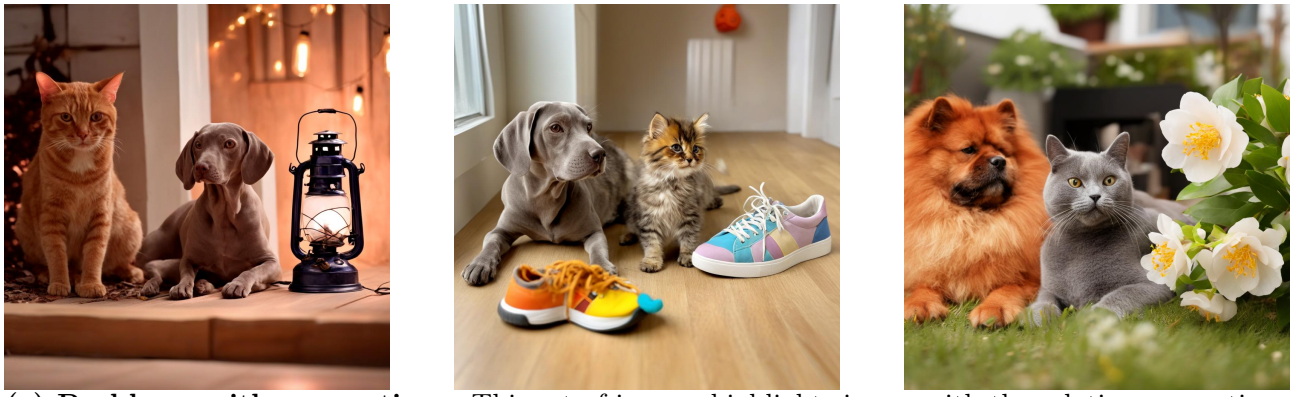
(a) Generated cap (top), original cap (bottom).

(b) Generated cup (top), original cup (bottom).

(c) Generated clock (top), original clock (bottom).

(d) Loss of fine-grained details. Certain fine but semantically relevant details are often misrepresented by the diffusion model. For example, the logo on the cap, originally a superimposed L and N, has become an A; the text *NeurIPS Conference* on the cup is now illegible; and the yellow number on the clock face has disappeared. While the overall structure of these objects remains recognizable, close inspection reveals inaccuracies.

(e)



(e) Problems with proportions. This set of images highlights issues with the relative proportions of the depicted target concepts. In the first image, the dog appears smaller than the cat; in the second, the shoe is the same size as the dog; and in the last image, the dog is as large as the cat. These proportions do not reflect those in the original reference images.

Figure 3.5

(f)

(f) Examples of generation failures when attempting to compose four personalized subjects. This set of images highlights issues with the generation of images with more than 3 personalized concept. In the first image, four subjects are depicted, but only the first dog matches its intended identity. The prompt specified a personalized cat followed by three distinct personalized dogs; however, the generated cat only loosely resembles the target, and two of the dogs appear similar but do not correspond to the intended identities. In the second image, the prompt included a cat, a grey dog, a pair of blue shorts, and a grey sloth plush toy. While visually similar objects are present, none faithfully reproduce the original personalized subjects. In the third image, the intended subjects were a teddy bear, a dog, a cat, and a clock. Only the teddy bear and dog are partially represented, with the cat and dog features erroneously fused, and the clock entirely missing.

Figure 3.6

Chapter 4

Description of the Experimental Setup

4.1 Introduction

This chapter describes the experimental procedure used to evaluate existing models on selected datasets, as part of this study.
The models considered during these experiments are the principal personalization models in literature: Yo'LLaVA (versions S and M) [8], RAP [12], and MC-LLaVA [10]. These models were evaluated on three main datasets: the Yo'LLaVA dataset, the MC-LLaVA dataset, and the novel MP-Bench dataset. In general, better performance is expected on the generated dataset, as its images tend to be simpler than the real photographs.
As summarized in Table 4.1, the experiments can be divided into two categories: those involving images with multiple personalized subjects, and those involving images with a single personalized subject.
The considered methods differ considerably in their capacity to personalize multiple concepts and in their scalability. RAP can scale to a maximum of two subjects per image. Yo'LLaVA-S operates exclusively on single concepts by design, while Yo'LLaVA-M is capable of incorporating multiple subjects, although its performance tends to degrade slightly as the number of concepts increases. MC-LLaVA is specifically designed for multi-subject personalization and, when trained appropriately, can scale to any number of subjects m.
The remainder of this chapter is organised as follows. In the first section, a detailed description of the datasets used in the experiments is provided. This is followed by an explanation of the tasks performed on the available dataset. The evaluation metrics are then introduced, followed by a description of the technical protocols employed during training. An in-depth overview of MN5, the high-performance computing cluster that hosted the experiments for this project, is provided in the appendix of this work.

4.2 Experimental Setup

4.2.1 Datasets

The experiments were conducted using three distinct datasets: two publicly available datasets introduced in Yo'llava and MC-LLaVA and MP-Bench, generated with MS-Diffusion.
The Yo'llava dataset consists of 40 diverse categories, including objects, buildings, and people.
For the "person" category, images were collected from micro-influencers (e.g., TikTok creators). Pets featured in the dataset are all personal pets of the authors of the paper, while landmarks are selected from local, less-known locations. Object images were sourced from Amazon product reviews.
To ensure the novelty of the subjects and eliminate prior model exposure, all categories underwent filtering using LLaVA [16], which was queried with *"Do you know this person/subject?"*. Only subjects for which LLaVA returned a negative or ambiguous response were retained.

Dataset	Concepts	Samples	Multi-Subject support
Yo'llava	1	333	×
MC-LLaVA	1	744	✓
	2	270	
	3–4	90	
MP-Bench	1	325	✓
	2	111	
	3	176	

Table 4.1: Summary of dataset characteristics: number of personalized concepts per image, number of test samples per setting, and support for multi-subject personalization.

Each category in the dataset provides approximately 5 to 10 images for training and an additional 5 to 10 for testing. All evaluations with this dataset are conducted under the single-subject setting; multi-subject personalization is not supported by Yo'llava dataset.

The MC-LLaVA dataset offers a more complex evaluation setup. It comprises a training set of single-concept images and a test set that includes both single- and multi-concept images.
The dataset contains around 2000 images in total, with 10 images per category in the training set. The dataset comprises two splits: one with 33 concepts, where test images combine 3 to 4 concepts, and another with 60 concepts, where test images combine 2 concepts. The categories span a wide range of visual concepts, including cartoon animals (18), human actors (34), humans in sports contexts (3), manga characters (35), as well as cartoon characters (2) and 3D action figures (3). This diversity makes the dataset well-suited for evaluating models' compositional reasoning capabilities and their generalization to multi-concept.

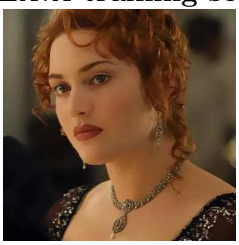
Finally, MP-Bench dataset was generated to simulate structured and controlled concept compositions. This generated dataset is divided into two splits. The first split is derived from MS-Bench and includes synthetic subjects designed for single-image and triple-subject personalization scenarios. The second split is based on data derived from Yo'llava and supports tests involving single-image as well as dual-subject personalization.

Figure 4.1 illustrates representative examples from the various datasets employed in this study. The first row displays images from the MC-LLaVA training set, which primarily consists of single-concept instances. The second row includes test samples featuring both dual- and triple-target concept settings. The third row presents examples drawn from the MS-Bench training corpus. In the fourth row, selected images from the Yo'llava training set are shown. Finally, the fifth row highlights data from the newly introduced MP-Bench, specifically showcasing samples from its multi-concept test split.

Examples from MC-LLava training set


⟨Harry⟩


⟨HouTeng⟩


⟨Rose⟩


⟨Xue⟩

Examples from MC-LLava test set with multiple concepts


⟨Alex ⟩, ⟨Gloria ⟩, ⟨Marti ⟩

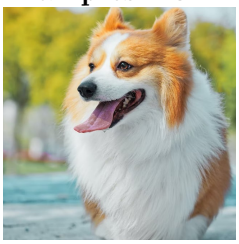

⟨HouTeng ⟩, ⟨HongXia ⟩

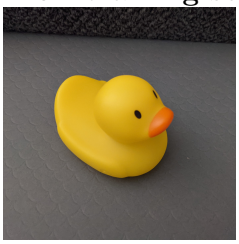

⟨Jack ⟩, ⟨Rose ⟩


⟨Harry ⟩, ⟨Hermione ⟩, ⟨Rohn ⟩

Examples from MS-Bench training set

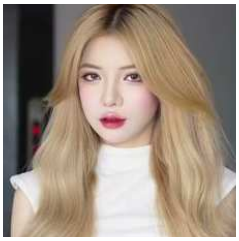

⟨cat1 ⟩


⟨dog ⟩


⟨ducktoy ⟩


⟨teapot ⟩

Examples from Yo'LLaVA training set


⟨ciin ⟩


⟨cat-cup ⟩


⟨shiba-yellow ⟩


⟨denisdang ⟩

Examples from MP-Bench test set with multiple concepts


⟨bo ⟩, ⟨butin ⟩


⟨cat1 ⟩, ⟨dog6 ⟩, ⟨ducktoy ⟩


⟨dog5 ⟩, ⟨teddybear ⟩, ⟨coat ⟩

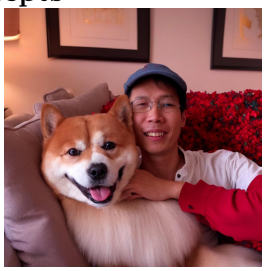

⟨bo ⟩, ⟨oong ⟩

Figure 4.1: Sample images from the four datasets used in this work: MC-LLaVA , MS-Bench, Yo'llava, and MP-Bench. The datasets include both single- and multi-concept personalized images across diverse scenes and categories.

4.2.2 Tasks

The numerical experiments conducted in this work aims to assess the recognition and captioning capabilities of the different personalization algorithms under consideration.

Recognition Experiments. The recognition capabilities of three models, MC-LLaVA , Yo'LLaVA, and RAP, are evaluated across three datasets: MP-Bench dataset, the MC-LLaVA dataset, and the Yo'LLaVA dataset. The number of target concepts, denoted by C, varies across datasets: $C = 1, 3$ for MP-Bench dataset; $C = 1, \ldots, 4$ for the MC-LLaVA dataset; and $C = 1$ for the Yo'LLaVA dataset which does not support multi-concept personalization.
The evaluation protocol, adapted from the MC-LLaVA testing methodology, is consistently applied across all models. The following symbols are used throughout this section:

- C: number of target concepts present in the image.
- T: number of test images per concept in the single-concept setting (assumed constant across concepts).
- M: number of multi-concept test scenarios.
- T_m: number of test images per multi-concept setting (assumed constant).
- m: number of concepts present in a single multi-concept image (with $m \leq C$).

In the single-concept setting, each of the C target concepts is associated with T test images containing only that concept, yielding a total of $C \times T$ test images. For recognition task, each image is queried with all C concepts, resulting in $C \times T$ positive samples (each image correctly matched with its concept) and $C \times T \times (C-1)$ negative samples (each image queried with all incorrect concepts).
In addition, 50 external single-concept images that do not contain any of the C concepts are randomly sampled. Each of these is queried with all C concepts, generating an additional $50 \times C$ negative samples.
For the multi-concept setting, M scenarios were considered. Each setting consists of T_m test images, where each image contains up to m concepts (with $m \leq C$). Every image is queried for the presence of each individual concept and for the complete concept set of lenght m, yielding up to $M \times (T_m \times m + T_m) = M \times T_m \times (m+1)$ positive samples.
In addition, 50 external multi-concept images not containing any of the target concepts are selected. These are queried with several of the C concepts, contributing with a maximum of $50 \times C$ further negative samples.
Combining both settings, the total number of recognition queries is:

- Positive samples: $\underbrace{C \times T}_{\textbf{single-concept}} + \underbrace{M \times T_m \times (m+1)}_{\textbf{multi-concept}}$
- Negative samples: $\underbrace{C \times T \times (C-1)}_{\textbf{single-concept}} + \underbrace{50 \times C}_{\textbf{external single-concept}} + \underbrace{50 \times C}_{\textbf{external multi-concept}}$

All test queries follow a unified question template:

> *"Can you see <$concept_1$ ... $concept_i$ ... $concept_m$> in this photo? Answer the question using a single word Yes or No."*

Here, the argument of the parethesis $<\ldots>$ denotes either a single target concept or a combination of concepts in multi-concept scenarios.

Captioning Experiments. For the captioning experiments, all test images in the C-concept setting are considered, encompassing both single- and multi-concept images. The number of test images for each dataset and configuration is reported in Table 4.1.
The model is prompted to generate a caption for each image, and its performance is evaluated by

verifying the presence of identifiers corresponding to the visual concepts depicted in the image.
This evaluation provides insight into the model's ability to accurately describe both isolated and co-occurring concepts in complex visual scenes.
Captioning is performed exclusively during the test phase, using a single unified prompt for both single- and multi-concept images. We consider two protocols for captioning:

- MC-LLaVA -derived protocol:

  "Can you see <concept_1>...<concept_m> in the image? Don't answer the question, but remember it, and only respond with a detailed caption for the image. Your caption:"

  This prompt introduces bias by explicitly priming the model with the target concepts, which can influence the generated caption.

- RAP-derived protocol:

  "Please provide a caption for this image."

  This neutral prompt avoids any concept priming, allowing for a more genuine assessment of the model's ability to personalize captions without relying on predefined target concepts.

It is important to note that all compared methods, except RAP, require access to the ground-truth target concepts at test time in order to load the corresponding concept embeddings and language model heads. This reliance represents a significant limitation. In contrast, RAP does not require the user to know the ground-truth concepts at inference time, making it more flexible and applicable in real-world scenarios.

4.2.3 Evaluation Metrics

To assess recognition performance, standard classification metrics were employed: accuracy, precision recall and F1-score. Accuracy measures the proportion of correct classifications, both positive and negative, and is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

In this context:

- TP (True Positives): the number of times the model correctly predicts the presence of a target concept (i.e., it answers *yes* when the concept is actually present in the image);
- TN (True Negatives): the number of times the model correctly predicts the absence of a target concept (i.e., it answers *no* when the concept is not present);
- FP (False Positives): the number of incorrect *yes* responses when the concept is actually absent;
- FN (False Negatives): the number of incorrect *no* responses when the concept is actually present.

In our recognition framework, a positive classification occurs when the model correctly answers *yes* to the presence of a target concept in the image; a negative classification corresponds to a *no* response when the concept is not present.
Due to the inherent imbalance in the considered testing protocol, where negative queries significantly outnumber positive ones, accuracy alone may not provide a comprehensive performance assessment. For this reason, recall and precision were computed.
Recall, also known as the true positive rate (TPR), measures the model's ability to identify all relevant instances and is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Within the considered recognition setting, recall reflects the fraction of images in which all target concepts are correctly identified as present. In highly imbalanced datasets, recall is a more informative metric than accuracy, as it captures the model's effectiveness in retrieving true positives.

Precision evaluates the proportion of positive classifications made by the model that are actually correct. It is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

While precision increases when false positives are reduced, recall improves with fewer false negatives. Both metrics offer complementary insights into model performance.
Finally, the F1-score, defined as the harmonic mean of precision and recall, offers a balanced evaluation:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

To evaluate captioning performance, the following protocol was adopted. From each model-generated caption, the personalized concept identifiers, which are easily identifiable by their angular brackets (e.g., `<dog>` or `<hat>`) were extracted. Then precision, recall, and the F1-score were computed.
In this context, recall measures the proportion of target concepts correctly mentioned in the caption, while precision quantifies the fraction of correctly retrieved concepts over the total number of predicted concepts.

4.3 Technical details about the training

4.3.1 Masks generation for MC-LLaVA method

In order to initialize the concept tokens, MC-LLaVA method requires the extraction of visual tokens from concept images. This process involves the generation of masks for the training images and is performed offline. Notably, this procedure is specific to the MC-LLaVA method and is not required by the other two methods considered in this work.
For this project, masks were generated using the Segment Anything Model (SAM) [38], a well-established model for promptable image segmentation that has demonstrated impressive zero-shot performance. Given an imput image, SAM is capable of producing high-quality object masks from input prompts such as points, bounding boxes, or textual descriptions. In this study, point annotations are used to identify the target subject of the image.
To simplify the annotation process, the VGG Image Annotator (VIA) [39], a manual annotation tool, was employed. VIA enables the loading of a set of images, for example the training images corresponding to the $i-th$ concept, and the marking of regions containing the target subject using points or other shapes. The resulting manual annotations can be exported in plain text formats such as $JSON$. The raw annotations generated with VIA were subsequently processed and provided as input to SAM, paired with the corresponding images. The final output of this pipeline is a high-quality mask that delineates the target concept within each image.

4.3.2 Generation of Hard Negative Samples for Training

The Yo-LLaVA and MC-LLaVA methods require a training phase that incorporates hard negative samples. These consist of a set of images (approximately 100–150 per concept) containing subjects that are similar to the target concept but not identical. Such examples are intended to be challenging for the model to distinguish from positive samples, thereby encouraging the model to learn how to differentiate the target concept from visually similar subjects belonging to the same class.
While the Yo-llava dataset includes negative samples for each target concept, the MC-LLaVA dataset, at the time of writing, did not provide corresponding negative samples. This made it necessary to design a dedicated pipeline for generating hard negatives.
An initial attempt relied on the use of Bing Image Downloader, a Python library that enables the bulk downloading of images from `Bing.com` based on a specified query string. The tool constructs a Bing Image Search URL based on the input query, filters, and settings; it then retrieves the `HTML` content of the search results, extracts image URLs via regular expressions, and downloads each image. The

downloader is capable of efficiently fetching and saving a large number of images within a short period, continuing the process until the specified limit is reached. In this baseline pipeline, a simple query string describing the target concept's class was provided as input.
However, this approach did not yield hard negative samples of sufficient quality for each target concept. In particular, it proved challenging to collect appropriate negative samples for concepts involving human subjects or cartoon characters.
To address these limitations, the pipeline was enhanced through two key improvements. First, a post-processing module based on CLIP was introduced. After downloading the raw images from Bing, each image was encoded into the CLIP embedding space and compared to the encoded original image of the target concept. Only the top-k most similar images were retained, while the remainder were discarded. Second, an additional set of approximately 20 images was injected into the pool of Bing images prior to CLIP encoding. These supplementary images were randomly sampled from two dedicated datasets: one consisting of anime faces images [40] and the other consisting of human face images from a dataset designed to ensure balanced representation across race, gender, and age groups [41]. This additional sampling step was performed only when the target subject was a human or anime character. These sources were selected because the original dataset features a significant number of categories involving human and anime characters. The combined approach helped achieve a sufficient number of high-quality hard negative samples for the dataset.

Chapter 5

Experimental Evaluation

5.1 Introduction

The opening section of this chapter presents a comparative analysis aimed at evaluating the feasibility and implications of using a generated dataset in place of real-world data. The objective is to assess the extent to which synthetic data can serve as a reliable substitute without compromising the validity of experimental results, and to measure its influence on model performance across standard evaluation metrics.
Following this preliminary investigation, results from the main numerical experiments introduced in the previous chapter are reported. The outcomes of the recognition task are discussed first, followed by those related to the captioning task. For each task, evaluation is carried out using standard performance metrics, namely: recall, precision, accuracy, and F1-score.

5.2 Preliminary Evaluation: Impact of Generated Data on Model Performance

A dedicated analysis was conducted on the generated dataset to assess the validity of using synthetic data as a substitute for real-world data and to quantify any potential variation in model performance resulting from this substitution. The study aims to determine whether generated datasets can be considered a reliable proxy in recognition tasks.
This evaluation was conducted in two phases, first using the RAP-MLLM model and then the Yo'LLaVA model. The tests were limited to single-concept images only, in order to isolate and fairly assess model performance in a controlled setting. Specifically, the evaluation compared model behavior on two datasets: the original single-concept test set from the Yo'LLaVA paper, and a synthetic version of this dataset generated using the MS-Diffusion model. The synthetic dataset was created by using the original test set as a seed, carefully reproducing the same subjects and maintaining the same single-concept structure. This ensured that both datasets shared equivalent characteristics, allowing a direct comparison. The goal of this experiment was to assess whether the models could perform equally well on real-world data and its synthetic counterpart, under identical recognition tasks and conditions.
For each class present in the datasets, accuracy results were aligned and compared individually. This per-class analysis allowed for a detailed investigation of how the transition from real to generated images affects recognition performance across different target concepts. As shown in Figure 5.1 and 5.2, the impact of generated data on model performance varies depending on both the method used and the object category.
In the case of Yo'LLaVA (Figure 5.1), a noticeable performance drop is observed for certain concepts, such as yixing and dog, where the generated data leads to significantly lower accuracy compared to the original. However, in several other classes like boat, car, and cell, the performance of the model trained on generated data remains close to that of the original, indicating that the synthetic images in

those cases are sufficiently representative for the recognition task.
On the other hand, the RAP-MLLM model (Figure 5.2) shows a more stable behavior across target concepts. The differences in accuracy between original and generated datasets are generally less pronounced, and for most concpets, the generated data achieves comparable results to the original. This suggests that the RAP-MLLM method is more robust to the domain shift introduced by the synthetic data.
In conclusion, while generated data can serve as a reasonable proxy for real data in certain contexts, its effectiveness is highly dependent on both the target class and the underlying model. Therefore, careful validation remains essential when substituting real images with synthetic ones for model training or evaluation. More extensive results are provided in appendix B.

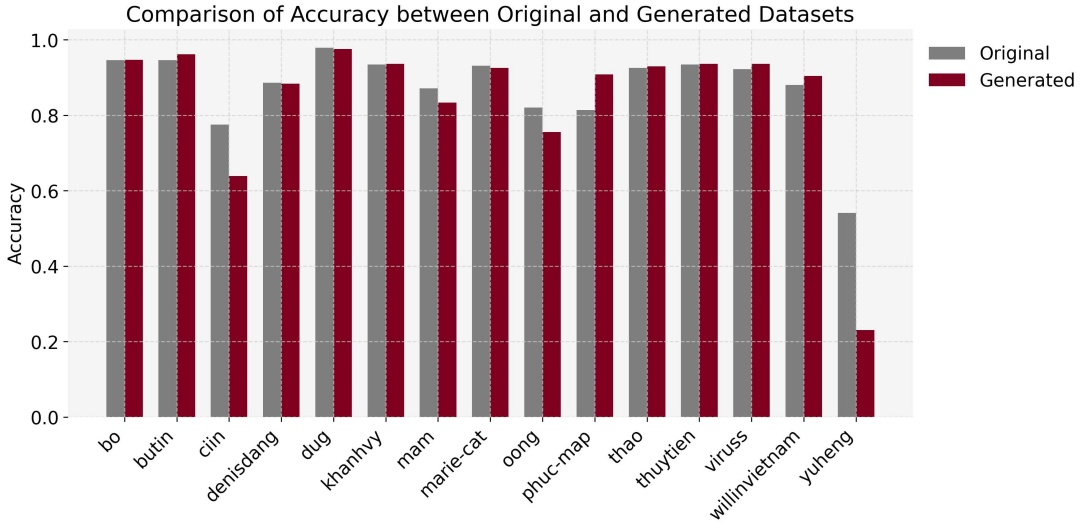


Figure 5.1: **Accuracy, Yo'LLaVA method.** Comparison of accuracy between the original Yo'LLaVA single-concept test set and a synthetic version generated from it using MS-Diffusion, broken down by target concepts. Results are obtained using the Yo'LLaVA method.

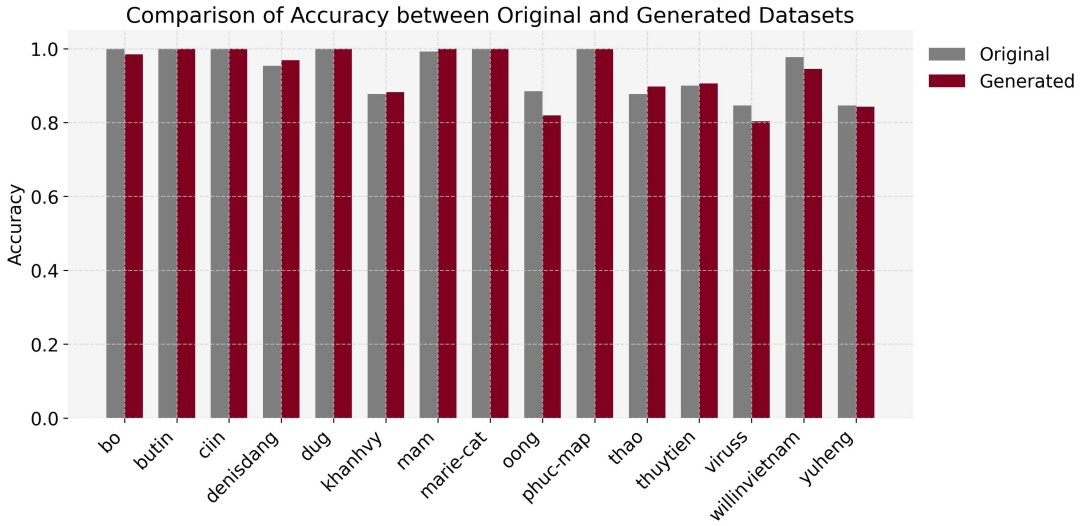


Figure 5.2: **Accuracy, RAP-MLLM method.** Comparison of accuracy between the original Yo'LLaVA single-concept test set and a synthetic version generated from it using MS-Diffusion, broken down by target concepts. Results are obtained using the RAP-MLLM method.

5.3 Recognition

This section presents the results of the numerical experiments conducted for the recognition task. All experiments were performed following the MC-LLaVA protocol, detailed in section 4.2.2. Some cells in the result tables are left blank due to limitations in the Yo'LLaVA dataset, which does not support testing on multi-subject images.
In the following, we present a series of tables that summarize the recognition accuracy results obtained in this study. The original recognition results reported in the MC-LLaVA paper are presented first, serving as a baseline for comparison. Following this the overall accuracy is presented, calculated as a weighted average of the results across two multi-subject configurations used in both the MC-LLaVA and the generated datasets. Subsequently, a breakdown of the accuracy per split is provided for both datasets.
For completeness, precision and recall values are also provided for all the methods and datasets. These metrics are essential for a more comprehensive evaluation, particularly in scenarios with class imbalance between positive and negative samples. Indeed, it is important to note that in the recognition task the distribution of positive and negative question–answer pairs is highly unbalanced. As detailed in the evaluation protocol in Section 4.2.2, the number of negative samples at inference time significantly exceeds the number of positive ones. In particular, the ratio between positive and negative instances is approximately 1 to 100, depending on the dataset under consideration. This imbalance highlights the fact that accuracy alone is not a sufficient metric for evaluating model performance in this task. In practice, a model may achieve high accuracy simply by correctly predicting the abundant soft negative cases, which are generally easier to handle. As a result, strong performance on these instances does not necessarily reflect true recognition capability or the model's ability to make fine-grained distinctions. For this reason, additional metrics such as precision and recall are essential to obtain a more comprehensive and reliable assessment.
The first Table 5.1 reports the recognition accuracy results as presented in the original MC-LLaVA paper. These serve as a baseline for comparison with the numerical results obtained in the present study, which are shown in the subsequent tables.
As summarized in Table 5.1, the Yo'LLaVA method demonstrates strong performance on both the MC-LLaVA single-concept split and the Yo'LLaVA dataset. Conversely, the Yo'llava-M variant (section 2.1.2) exhibits reduced performance in both single- and multi-concept settings, likely due to confusion arising from the presence of multiple visual concepts within a single image. The RAP-MLLM approach, which incorporates additional recognition modules and supports multi-concept queries via a top-K selection mechanism, achieves competitive accuracy results. The MC-LLaVA method, proposed in the referenced paper, achieves state-of-the-art recognition performance in both single- and multi-concept scenarios. However, the original work does not report results in terms of precision and recall, which limits the completeness of the evaluation, particularly in cases involving class imbalance.

Dataset	Yo'LLaVA-S			Yo'LLaVA-M			RAP			MC-LLaVA		
	single	multi	weighted	single	multi	weighted	single	multi	weighted	single	multi	weighted
mc-llava	0.84	-	0.84	0.74	0.73	0.74	0.75	0.69	0.71	0.91	0.85	0.88
yo'llava	0.92	-	0.92	0.92	-	0.92	0.85	-	0.85	0.95	-	0.95

Table 5.1: **Baseline Recognition Accuracy as Reported in the MC-LLaVA Study**. This table presents recognition accuracy results as reported in the original MC-LLaVA paper. The datasets evaluated are listed in the vertical axis and include both Yo'LLaVA and MC-LLaVA . The horizontal axis indicates the three evaluation methods used in the study, allowing direct comparison of performance across different protocols.

Tables 5.2 and 5.3 present the recognition results in terms of accuracy, obtained from the numerical experiments conducted as part of this study.
Specifically, Table 5.2 shows the accuracy across different datasets, with single- and multi-concept splits reported separately. This separation allows for a more precise evaluation of performance on test splits containing two and three visual concepts per image, and facilitates a direct comparison between methods under varying levels of complexity.
For the Yo'LLaVA-S method, performance across the different single-concept test datasets ranges from a minimum of 0.74 on the Yo'LLaVA dataset to a maximum of 0.85 on the single-concept split of MP-Bench dataset. The latter is considered less challenging, as it does not include human subjects and, being synthetically generated, presents simpler visual structures overall.
In contrast, the performance of the Yo'LLaVA-M variant is notably lower than the results originally reported in the MC-LLaVA paper. This discrepancy is likely due to the absence of publicly available code at the time of experimentation. Consequently, the implementation used here was developed based on the descriptions and methodology outlined in the original paper, which may have introduced deviations.
The RAP-MLLM model achieves consistently high accuracy, exceeding 0.90 in single-concept recognition across all datasets. As expected, on the MC-LLaVA dataset performance tends to decrease as the number of concepts per image increases, specifically, from the multi-2 to the multi-3 configuration. In the generated dataset, the two-concept split (based on Yo'LLaVA) includes a high proportion of human faces and is therefore more challenging than the three-concept split derived from MS-Bench. This leads to slightly reduced accuracy on the Yo'LLaVA-derived split.
The MC-LLaVA model demonstrates strong overall performance; however, the results obtained in the present study are, on average, marginally lower than those reported in the original publication. This discrepancy is likely due to the unavailability of specific components used during the original training process, most notably, the official set of negative samples, which required the construction of a custom implementation pipeline. Furthermore, the complete dataset employed in the original work was not fully released at the time of experimentation, potentially contributing to the observed performance gap.

Dataset	Yo'LLaVA-S			Yo'LLaVA-M			RAP			MC-LLaVA		
	single	multi	weighted	single	multi	weighted	single	multi	weighted	single	multi	weighted
MP-Bench ms-bench	0.85	-	0.85	-	0.35	0.35	0.97	0.88	0.93	0.77	0.81	0.79
MP-Bench yo'llava	0.74	-	0.74	-	0.33	0.33	0.95	0.80	0.88	0.73	0.75	0.74
mc-llava multi 3	0.75	-	0.75	-	0.53	0.53	0.93	0.75	0.88	0.81	0.85	0.82
mc-llava multi 2	0.80	-	0.80	-	0.26	0.26	0.90	0.93	0.91	0.80	0.80	0.80
yo'llava	0.74	-	0.74	-	-	-	0.94	-	0.94	0.73	-	0.73

Table 5.2: **Detailed Accuracy Results Across Dataset Splits**. This table reports the accuracy results obtained through the conducted recognition experiments. The evaluation was performed across different dataset configurations, listed on the vertical axis, while the horizontal axis indicates the various tested methods. The results reflect the performance of each method on both the MC-LLaVA and MP-Bench datasets, with separate entries for each dataset split considered during evaluation. For each dataset, the best performance on single-concept test images and on multi-concept test images is highlighted in the table.

Table 5.3 reports the same accuracy results, this time aggregated as weighted averages across the dataset splits of both the generated and MC-LLaVA datasets. Specifically, for each dataset, the two single-concept splits were averaged by assigning greater weight to the bigger split. A similar approach was applied to the multi-concept configurations: the results for the two- and three-concept splits were combined into a single weighted average, reported under the column labeled *multi*. This formulation

allows for a more compact representation of the performance.

Dataset	Yo'LLaVA-S			Yo'LLaVA-M			RAP			MC-LLaVA		
	single	multi	weighted	single	multi	weighted	single	multi	weighted	single	multi	weighted
MP-Bench	0.85	-	0.85	-	0.35	0.35	0.96	0.86	0.91	0.88	0.79	0.88
mc-llava	0.78	-	0.78	-	0.35	0.35	0.92	0.87	0.90	0.83	0.81	0.83
yo'llava	0.74	-	0.74	-	-	-	0.90	-	0.90	0.79	-	0.79

Table 5.3: **Weighted Accuracy Results Across Dataset Splits**. This table presents the overall recognition accuracy obtained by weighting the results across the different dataset splits. The values are computed as weighted averages of the accuracy scores from each split of both the MC-LLaVA and MP-Bench datasets. The averaging was performed in two stages: first, results from single-concept tests and multi-concept tests were averaged separately. Within each category, results from the respective splits were weighted according to their relative number of samples, namely, split a and split b for the MC-LLaVA dataset, and MS-Bench and Yo'LLaVA for the generated dataset (MP-Bench). Subsequently, for each dataset, the single-concept and multi-concept averages were combined to compute the final value reported in the *weighted* column. Dataset names are listed on the vertical axis, while the tested methods are shown along the horizontal axis. For each dataset, the best performance on single-concept test images and on multi-concept test images is highlighted in the table.

Tables 5.4 and 5.5 report the precision and recall results, respectively, across all four evaluated methods and for each dataset split.
Focusing first on precision, this metric reflects the proportion of predicted positive instances that are actually correct, and it improves as the number of false positives decreases. As observed, the Yo'LLaVA-S method tends to frequently produce affirmative predictions, even when the ground truth label is negative. This behavior negatively affects precision and may be mitigated by increasing the number of negative training samples, thereby improving the model's ability to distinguish between relevant and irrelevant concepts.
Yo'LLaVA-M, which is derived from Yo'LLaVA-S and trained using the same protocol and the same set of positive and negative templates, exhibits similar issues with precision performance.
In contrast, the RAP-MLLM method demonstrates improved precision, particularly in scenarios involving multiple target concepts within a single scene. Likewise, MC-LLaVA outperforms both Yo'LLaVA variants in terms of precision and maintains stable performance even when transitioning from single- to multi-concept image splits.
Recall, which increases as the number of false negatives decreases, exhibits markedly different trends across the evaluated methods, as reported in Table 5.5. For the Yo'LLaVA-S model, recall values are on average around 0.6. This reflects the model's tendency to respond affirmatively to most queries, regardless of the actual presence of the target concept, resulting in fewer false negatives but a corresponding drop in precision.
Yo'LLaVA-M shows slightly improved recall, with values consistently around 0.7 across all datasets. In contrast, the RAP-MLLM method achieves strong recall performance, reaching values as high as 0.99 on the MC-LLaVA dataset. This indicates a strong ability to correctly detect the absence of target concepts and, crucially, to avoid missing relevant instances in the image.
The MC-LLaVA model, while competitive in terms of precision, shows relatively lower recall performance. This suggests a more conservative prediction behavior, likely favoring precision at the expense of occasionally missing relevant concepts.

Dataset	Yo'LLaVA-S			Yo'LLaVA-M			RAP			MC-LLaVA		
	single	multi	weighted	single	multi	weighted	single	multi	weighted	single	multi	weighted
MP-Bench ms-bench	0.13	-	0.13	-	0.070	0.070	0.66	0.98	0.81	0.26	0.22	0.24
MP-Bench yo'llava	0.047	-	0.047	-	0.11	0.11	0.48	0.88	0.66	0.51	0.40	0.46
mc-llava multi 3	0.11	-	0.11	-	0.39	0.39	0.26	0.82	0.40	0.58	0.43	0.54
mc-llava multi 2	0.12	-	0.12	-	0.087	0.087	0.26	0.94	0.50	0.41	0.44	0.42
yo'llava	0.047	-	0.047	-	-	-	0.68	-	0.68	0.51	-	0.51

Table 5.4: **Precision results obtained from the recognition experiments**. The table reports precision values across the tested methods (shown horizontally) and datasets (listed vertically), including different evaluation splits. For each dataset, the best performance on single-concept test images and on multi-concept test images is highlighted in the table.

Dataset	Yo'LLaVA-S			Yo'LLaVA-M			RAP			MC-LLaVA		
	single	multi	weighted	single	multi	weighted	single	multi	weighted	single	multi	weighted
MP-Bench ms-bench	0.41	-	0.41	-	0.71	0.71	0.96	0.91	0.94	0.11	0.19	0.15
MP-Bench yo'llava	0.58	-	0.58	-	0.72	0.72	0.94	0.91	0.92	0.19	0.26	0.22
mc-llava multi 3	0.65	-	0.65	-	0.74	0.74	0.98	0.84	0.94	0.27	0.26	0.26
mc-llava multi 2	0.69	-	0.69	-	0.72	0.72	0.99	0.98	0.98	0.11	0.34	0.17
yo'llava	0.58	-	0.58	-	-	-	0.96	-	0.96	0.19	-	0.19

Table 5.5: **Recall results obtained from the recognition experiments**. The table displays recall values for each tested method (columns) and dataset configuration (rows), covering all relevant datasets. For each dataset, the best performance on single-concept test images and on multi-concept test images is highlighted in the table.

5.4 Captioning

This section presents the results of the numerical experiments conducted for the captioning task. All experiments were carried out following the RAP-MLLM protocol, as described in Section 4.2.2.
A few preliminary considerations are necessary to properly interpret the results. First, the definition of the captioning task varies significantly depending on the model in question. In RAP-MLLM [12], captioning follows the standard definition, namely the task of describing the content of an image in natural language. Conversely, Yo'LLaVA [8] and MC-LLaVA [10] adopt alternative formulations of the captioning task, which significantly influence the final performance. In the case of Yo'LLaVA, the model is typically prompted to generate captions suitable for social media contexts (e.g., *Hey, can you see* `<concept1>` *and* `<concept2>` *in this photo? Could you write a cute and detailed Facebook caption for it?*). For MC-LLaVA , the prompt instead asks the model to recognize certain concepts but respond only with a caption (e.g., *Can you see* `<concept1>` ... `<conceptm>` *in the image? Don't answer the question, but remember it, and only respond with a detailed caption for the image. Your caption:*). In both cases, the input query explicitly mentions the target subjects of the image. This differs fundamentally from the RAP-MLLM protocol, which provides a more general and unconstrained prompt. The latter is considered more appropriate for evaluating generic image captioning capabilities, as it avoids injecting prior knowledge and better reflects the true complexity of the task. Therefore, this definition is adopted in the present work.
Second, due to the more rigorous and unbiased evaluation protocol employed in this study, the performance of Yo-llava and MC-LLaVA deteriorated significantly when assessed under these conditions, compared to results obtained using their original protocols. For this reason, their results are omitted.
The captioning results obtained using RAP-MLLM are reported in Tables 5.6, 5.7 and 5.8. The metrics shown include precision, recall and F1-score.
As shown by the results in Table 5.6, the precision of the captioning task using the RAP-MLLM model is high on datasets containing a single target concept. A slight deterioration is observed when moving to the multi-subject setting. This trend becomes more pronounced when considering the recall (Table 5.7) and F1-score (Table 5.8) metrics, where the performance drop from the single- to multi-subject setting is both substantial and clearly noticeable.
Qualitatively, this degradation is also reflected in the analysis of the raw outputs produced by the model. In the multi-concept setting, the generated captions often become overly generic and fail to explicitly mention any of the target concepts present in the image. In contrast, in the single-subject setting, the model is typically able to correctly identify the relevant concept and explicitly include its identifier in the generated caption.
The precision scores remain relatively high even in the multi-concept case, which may seem counterintuitive at first. This outcome is due to the model's low false positive rate: when RAP-MLLM does predict a specific concept in its caption, it is usually correct. In other words, although the model struggles to produce personalized or detailed captions in the multi-concept scenario, it rarely makes incorrect identifications.
The main issue arises from the opposite direction, namely a high number of false negatives. In the multi-subject setting, the model often fails to mention even a single relevant concept, instead defaulting to generic descriptions that do not address the image's specific content. This tendency to omit key concepts leads to a significant increase in false negative instances, in which the model fails to recognise and mention a concept that is in fact present. As a direct consequence, the recall metric, which measures the model's ability to retrieve all relevant instances, is notably low in the multi-concept setting.

Dataset	RAP		
	single	**multi**	**weighted**
MP-Bench ms-bench	**0.88**	0.70	0.82
MP-Bench yo'llava	0.74	0.70	0.72
mc-llava multi 3	0.72	0.75	0.73
mc-llava multi 2	0.71	**0.86**	0.76
yo'llava	0.82	-	0.82

Table 5.6: **Precision results obtained from the captioning experiments**. The table reports precision values across the tested method RAP-MLLM (shown horizontally) and datasets (listed vertically), including different evaluation splits. The best performance on single-concept test images and on multi-concept test images is highlighted in the table.

Dataset	RAP		
	single	**multi**	**weighted**
MP-Bench ms-bench	0.61	0.070	0.43
MP-Bench yo'llava	0.73	0.032	-
mc-llava multi 3	0.42	0.055	0.32
mc-llava multi 2	0.41	**0.21**	0.34
yo'llava	**0.73**	-	0.73

Table 5.7: **Recall results obtained from the captioning experiments**. The table reports recall values across the tested method RAP-MLLM (shown horizontally) and datasets (listed vertically), including different evaluation splits. The best performance on single-concept test images and on multi-concept test images is highlighted in the table.

Dataset	RAP		
	single	**multi**	**weighted**
MP-Bench msbench	0.72	0.12	0.32
MP-Bench yo'llava	0.77	0.060	-
mc-llava multi 3	0.53	0.10	0.42
mc-llava multi 2	0.52	**0.34**	0.46
yo'llava	**0.77**	-	0.77

Table 5.8: **F1-score results obtained from the captioning experiments**. The table reports F1-score values across the tested method RAP-MLLM (shown horizontally) and datasets (listed vertically), including different evaluation splits. The best performance on single-concept test images and on multi-concept test images is highlighted in the table.

To provide a concise overview of the results, a summarizing table is included. This table 5.9 presents aggregated metrics derived from the individual splits of the generated and MC-LLaVA datasets. Specifically, for each dataset, the metrics are averaged across the two single-subject splits and, separately, across the two multi-subject splits. These averages offer a representative summary of model performance over the entire dataset. The averaging process is weighted according to the number of samples in each split, in order to account for variations in data distribution.
In the case of the MC-LLaVA dataset, the most populous split, the one containing images with only two target concepts, also corresponds to the best performance, likely due to the reduced complexity of the visual scenes. Conversely, in the MP-Bench generated dataset, the two-concept split (Yo'LLaVA) appears to be the most challenging. This increased difficulty can be attributed to the frequent presence of human-related concepts, which tend to be highly similar and thus harder to distinguish accurately.

Dataset	Precision		Recall		F1-Score	
	Single	**Multi**	**Single**	**Multi**	**Single**	**Multi**
MP-Bench	0.824	0.700	0.586	0.0551	0.684	0.0968
mc-llava	0.710	0.820	0.420	0.150	0.520	0.250
yo'llava	0.820	-	0.730	-	0.770	-

Table 5.9: **Precision, Recall, and F1-score for single- and multi-concept captioning using the RAP-MLLM model.** Metrics are reported for the RAP-MLLM model on MP-Bench, MC-LLaVA , and Yo'LLaVA datasets. Values are computed separately for the single-subject and multi-subject splits. Within each split, metrics are weighted according to the number of images contained in each dataset subset, to account for differences in sample size across datasets.

5.5 Qualitative results

This section presents a selection of the most significant qualitative results related to the recognition and captioning tasks. Each example includes the input image, the query, and the corresponding output generated by the model. The samples comprise both successful instances, where the model accurately performs the downstream task, and failure cases that highlight limitations.
At this stage, the emphasis is placed on analyzing the behavior of different models, rather than comparing performance across datasets. The primary objective is to assess the extent to which the models are capable of handling the required tasks, as well as to identify their shortcomings.
The present section provides a partial and exclusively qualitative illustration of the models' behavior across the different tasks addressed in this study. The examples have been selected with the sole purpose of offering an indicative insight into the nature of the models' raw outputs. This analysis is inherently incomplete and should not be interpreted as a substitute for the quantitative evaluations reported in the previous sections, which are based on statistically significant test sets comprising hundreds or thousands of QA pairs per task.
The following analysis proceeds in two parts: first, selected examples of question-answer prediction pairs produced by each model are shown for the recognition task; subsequently, a similar comparative analysis is conducted for the captioning task.

5.5.1 Qualitative results: Recognition

This section presents a selection of representative results from the recognition task, highlighting performance differences across the models analyzed in this study.
Examples from a) to c) in Table 5.10 illustrate the recognition behavior of the MC-LLaVA model on the dataset introduced in the corresponding original paper. In sample a), the model correctly identifies the presence of both target concepts within the image. In contrast, in b) the model fails to recognize the absence of the queried concepts and produces two false positives, despite a marked visual dissimilarity between the actual characters and those it incorrectly predicts. In c), the model is queried about the presence of only one out of two concepts depicted in the image and provides a correct response.
Examples from d) to g) in Table 5.10 and Table 5.11 provide further insight into the capabilities of the RAP-MLLM model, which demonstrates to achieve the highest recognition accuracy in both single and multi-concept scenarios. In d), although the image contains three concepts, the model is queried about a single one and correctly confirms its presence. A similar query structure is applied in e), but in this case, the model returns an incorrect answer. Examples f) and g) involve negative queries, where the model is asked to confirm the presence of two target concepts that are not actually depicted. While RAP correctly responds in f) by identifying the mismatch, it fails in g) by incorrectly asserting the presence of the queried concepts.
Examples from h) to k) in Table 5.12 correspond to predictions by the Yo'LLaVA-M model on multi-concept images. Sample h) represents a positive query, where all mentioned concepts are indeed present and correctly recognized. In contrast, i) and k) depict negative query scenarios, where the model is expected to identify the absence of certain target concepts. While the model performs correctly in one case, it fails in the other, demonstrating variability in handling complex multi-entity scenes.
Finally, examples l) and m) in Table 5.12 and Table 5.13 showcase predictions from the Yo'LLaVA-S model on single-concept images under negative query conditions. In l), the model is queried about the presence of a character, ⟨butin ⟩, who does not appear in the image, which actually depicts a different character, ⟨viruss⟩; the model correctly responds *No*. In m), a random image, sourced from a control set not containing any personalized concepts, is used to test the model's ability to reject spurious matches. When asked about the presence of ⟨marie-cat⟩, the model again correctly responds negatively, indicating robustness in the presence of unseen or irrelevant visual content.

a)		**Query:** Can you see ⟨Tongren ⟩ and ⟨Yasina ⟩ in this photo? **GT Concepts:** Yes (⟨Tongren ⟩, ⟨Yasina ⟩) **Model output:** Yes **Dataset:** MC-LLaVA **Method:** MC-LLaVA
b)		**Query:** Can you see ⟨Tongren ⟩ and ⟨Yasina ⟩ in this photo? **GT Concepts:** ⟨Shaoyu ⟩, ⟨Tianming ⟩ **Model output:** Yes **Dataset:** MC-LLaVA **Method:** MC-LLaVA
c)		**Query:** Can you see ⟨AnXin ⟩ in this photo? **GT Concepts:** ⟨GaoQiQiang ⟩, ⟨AnXin ⟩ **Model output:** Yes **Dataset:** MC-LLaVA **Method:** MC-LLaVA
d)		**Query:** Can you see ⟨jiubuliulang ⟩ in the image? **GT Concepts:** ⟨zhongtangxi ⟩, ⟨sanchengmeiqin ⟩, ⟨jiubuliulang ⟩, ⟨donghailinxizi⟩ **Model output:** Yes **Dataset:** MC-LLaVA **Method:** RAP-MLLM

Table 5.10: Qualitative examples of QA pairs for the recognition task across different models and datasets. Each example includes the input image, the query regarding the presence of specific target concepts, the ground truth answer, and the corresponding model prediction.

e)		**Query:** Can you see ⟨donghailinxizi ⟩ in the image? **GT Concepts:** ⟨zhongtangxi ⟩, ⟨sanchengmeiqin ⟩, ⟨jiubuliulang ⟩, ⟨donghailinxizi ⟩ **Model output:** No **Dataset:** MC-LLaVA **Method:** RAP-MLLM
f)		**Query:** Can you see ⟨ciin ⟩ and ⟨dug ⟩ in the image? **GT Concepts:** ⟨thuytien ⟩, ⟨yuheng ⟩ **Model output:** No **Dataset:** MP-Bench **Method:** RAP-MLLM
g)		**Query:** Can you see ⟨bo ⟩ and ⟨viruss ⟩ in the image? **GT Concepts:** ⟨marie-cat ⟩, ⟨dug ⟩ **Model output:** Yes **Dataset:** MP-Bench **Method:** RAP-MLLM
h)		**Query:** Can you see ⟨Alex ⟩, ⟨Gloria ⟩ and ⟨Marty ⟩ in this photo? **GT Concepts:** ⟨Alex ⟩, ⟨Gloria ⟩, ⟨Marty ⟩ **Model output:** Yes **Dataset:** MC-LLaVA **Method:** Yo'LLaVA-M

Table 5.11: Qualitative examples of QA pairs for the recognition task across different models and datasets. Each example includes the input image, the query regarding the presence of specific target concepts, the ground truth answer, and the corresponding model prediction.

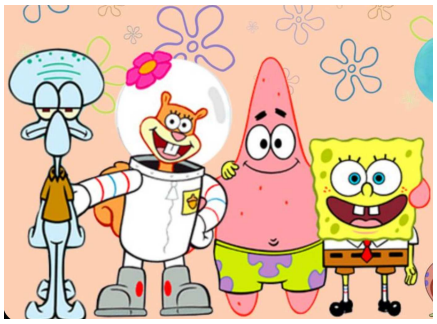
i)		**Query:** Can you see ⟨Alex ⟩, ⟨Gloria ⟩ and ⟨Marty ⟩ in this photo? **GT Concepts:** ⟨haimianbaobao ⟩, ⟨paidaxing ⟩, ⟨zhangyuge ⟩, ⟨shandi ⟩ **Model output:** Yes **Dataset:** MC-LLaVA **Method:** Yo'LLaVA-M
j)		**Query:** Can you see ⟨jierui ⟩ in this photo? **GT Concepts:** ⟨tangmu ⟩, ⟨jierui ⟩ **Model output:** No **Dataset:** MC-LLaVA **Method:** Yo'LLaVA-M
k)		**Query:** Can you see ⟨tangmu ⟩ and ⟨jierui ⟩ in this photo? **GT Concepts:** ⟨datong ⟩, ⟨kaiqi ⟩ **Model output:** No **Dataset:** MC-LLaVA **Method:** Yo'LLaVA-M
l)		**Query:** Can you see ⟨butin ⟩ in this photo? **GT Concepts:** ⟨viruss ⟩ **Model output:** No **Dataset:** Yo'LLaVA **Method:** Yo'LLaVA-S

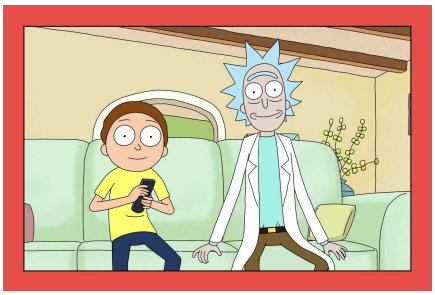
Table 5.12: Qualitative examples of QA pairs for the recognition task across different models and datasets. Each example includes the input image, the query regarding the presence of specific target concepts, the ground truth answer, and the corresponding model prediction.

m)		**Query:** Can you see ⟨marie-cat ⟩ in this photo? **GT Concepts:** No **Model output:** No **Dataset:** Random image **Method:** Yo'LLaVA-S

Table 5.13: Qualitative example of QA pair for the recognition task and Yo'LLaVA-S model on a soft negative image. The example includes the input image, the query regarding the presence of specific target concepts, the ground truth answer, and the corresponding model prediction.

5.5.2 Qualitative results: Captioning

This section presents a selection of results from the captioning task. In the following table 5.14 rows a) to c) are representative examples of the captioning performance of the MC-LLaVA method on the MC-LLaVA dataset. As can be observed from the comparison between the ground truth target concepts and the model predictions, the outputs tend to be generic descriptions of the images. In some cases, the model appears to identify characters as figures from films already present in the pretrained model's knowledge base. However, it consistently fails to explicitly name all the target concepts depicted in the images.
Conversely, rows d) to h) illustrate the performance of the RAP-MLLM model on the captioning task. In particular, rows d) to f) demonstrate successful cases in which the model produces complete and accurate captions, explicitly mentioning all the target concepts and describing their roles or actions within the scene. In contrast, rows g and h (evaluated on the MC-LLaVA multi 2 and multi 3 test splits) highlight failure cases. In row g), the model identifies only one of the three target concepts, yielding one true positive —Zhuyun— and two false negatives —Lixun and Laogao—. In row h), the model fails to recognize any of the target concepts, resulting in a generic caption and two false negatives.

a)		**Query:** Please, provide a caption for this image. **GT Concepts:** ⟨Modi ⟩, ⟨Rike⟩ **Model output:** Rick and Morty sitting on a couch. **Dataset:** MC-LLaVA **Method:** MC-LLaVA
b)		**Query:** Please, provide a caption for this image. **GT Concepts:** ⟨Jack⟩, ⟨Rose⟩ **Model output:** Two characters in a scene from a movie, sharing a moment of connection. **Dataset:** MC-LLaVA **Method:** MC-LLaVA

c)

Query: Please, provide a caption for this image.
GT Concepts: ⟨haimianbaobao⟩, ⟨paidaxing⟩, ⟨zhangyuge⟩, ⟨shandi⟩
Model output: SpongeBob and friends enjoying a sunny day at the beach.
Dataset: MC-LLaVA
Method: MC-LLaVA

---

d)

Query: Please, provide a caption for this image.
GT Concepts: <dog>
Model output: <dog> is a happy fluffy corgi dog.
He is sitting on a brick path with trees in the background.
<dog> looks like he is ready for a walk.
Dataset: MS-Bench
Method: RAP-MLLM

---

e)

Query: Please, provide a caption for this image.
GT Concepts: <dog8>
Model output: <dog8> is a happy pup!
He's enjoying a run in the woods.
Dataset: MS-Bench
Method: RAP-MLLM

---

f)

Query: Please, provide a caption for this image.
GT Concepts: <Modi>, <Rike>
Model output: <Rike> looks on with concern as <Modi > holds a shovel.
Dataset: MC-LLaVA
Method: RAP-MLLM

g)		**Query:** Please, provide a caption for this image. **GT Concepts:** <Lixun>, <Zhuyun>, <Laogao> **Model output:** <Zhuyun> is standing on a stage with two other people. **Dataset:** MC-LLaVA **Method:** RAP-MLLM
h)		**Query:** Please, provide a caption for this image. **GT Concepts:** <Shaoyu>, <Tianming> **Model output:** A scene from a video game featuring two characters, one with a purple outfit and the other with a green and yellow outfit. **Dataset:** MC-LLaVA **Method:** RAP-MLLM

Table 5.14: Qualitative example of QA pair for the recognition task and Yo'LLaVA-S model on a soft negative image. The example includes the input image, the query regarding the presence of specific target concepts, the ground truth answer, and the corresponding model prediction.

Chapter 6

Conclusions and perspectives

In this thesis, we have explored the challenge of multi-concept personalization in the realm of personalization of vision and language models.
Chapter 1 laid the groundwork by defining personalization within VLMs, emphasizing its motivations and practical applications. We reviewed relevant literature and identified the key difficulty of personalizing multiple target concepts within a single visual scene.
In Chapter 2, we surveyed existing personalization methods. We focused on training-based approaches including MyVLM, Yo'LLaVA, and MC-LLaVA. MyVLM introduces an external concept head to identify target concepts in images, combined with an embedding module that learns new representations for each concept within the intermediate feature space of the model. Yo'LLaVA instead learns new concept and descriptive token embeddings for each concept, training only these along with the associated classifier weights. MC-LLaVA builds on Yo'LLaVA, introducing a multi-concept instruction tuning strategy to better handle multiple concepts in a single scene.
We also reviewed training-free methods, in particular RAP, which relies on a database of concept-tagged images and short descriptions. RAP's pipeline enables the model to identify up to two target concepts in a test image and perform downstream tasks accordingly.
Chapter 3 explored personalized text-to-image generation, used to create a novel dataset, MP-Bench, supporting both single- and multi-concept tasks. We focused on diffusion-based methods, particularly the MS-Diffusion model, which supports multi-subject generation via the grounding resampler and multi-subject cross-attention mechanisms. We described the dataset construction process, including generation, cleaning, and qualitative evaluation (Sections 3.5.1–3.5.5). The final dataset consists of two splits, containing test images with one, two, or three personalized concepts. Limitations of the approach, as discussed in Section 3.5.5, include difficulties in rendering fine anatomical details (such as hands), maintaining realistic proportions, and handling compositions involving four or more personalized subjects.
Chapter 4 detailed the experimental setup, including the datasets used (our generated dataset MP-Bench, Yo'LLaVA dataset, and MC-LLaVA dataset), the downstream tasks (recognition and captioning), and the evaluation metrics (Sections 4.1–4.2).
In Chapter 5, we presented and analyzed experimental results. Preliminary evaluations showed that our synthetic dataset can effectively substitute real-world data without introducing bias.
Recognition results (Section 5.3) indicated that while models like RAP and Yo'LLaVA perform reasonably well in single-concept tasks, they often struggle in multi-subject settings. MC-LLaVA, designed specifically for multi-concept personalization, showed stronger performance in these scenarios.
Captioning results (Section 5.4) revealed that RAP was the only model among those tested to produce meaningful captions. However, it frequently failed to identify all present personalized concepts in multi-concept images, leading to many false negatives and lower recall. The other models performed poorly in captioning tasks, and their quantitative results were omitted for this reason.

These findings suggest promising directions for future research. In particular, training-free approaches such as RAP offer advantages in terms of scalability and usability. Unlike training-based models—such as MyVLM, which requires two separate training phases, or Yo'LLaVA and MC-LLaVA, which require training each concept individually (with each concept taking about an hour on an H100 GPU), training-free methods can scale to new concepts quickly.
Moreover, the simplicity of training-free models, combined with competitive (or sometimes superior) downstream performance, makes them strong candidates for practical deployment as personalized assistants. Their improved scalability and zero-training requirements make this line of research in personalization particularly promising for real-world applications.

Appendix A

Computational Environment

All experiments for this work were carried out on MareNostrum 5, a pre-exascale supercomputer of the EuroHPC initiative, located in Barcelona and hosted by the Barcelona Supercomputing Center (BSC). The majority of computations were performed on the Accelerated Partition (ACC), designed for GPU-intensive workloads. Job scheduling and resource allocation were managed using the Slurm Workload Manager, in accordance with standard practices in high-performance computing. To enable the concurrent execution of multiple algorithms, Slurm job arrays were utilized through the directive $array = \{indexes\}$, allowing for efficient batch processing of jobs with identical configurations. The use of HPC infrastructure was necessitated by the limited GPU memory of local computational resources, which proved insufficient for executing large-scale vision-language and multimodal models such as BLIP-2, Stable Diffusion XL, and LLaVA.

Appendix B

Impact of Generated Data on Model Performance

This appendix provides a more detailed analysis of the recognition experiments described in the main text, focusing specifically on the comparison between the original Yo'llava single-concept test set and its generated counterpart. In addition to accuracy results already discussed in the main body of the thesis, the evaluation includes precision and recall, allowing for a more comprehensive assessment of the models' performance.
The experiments were conducted using the same recognition task and constrained to the case of a single target concept per image. Both the RAP-MLLM and Yo'llava models were evaluated under identical conditions on the original and generated datasets. Results are reported per class to facilitate a fine-grained understanding of the models' behavior across different object categories.
Figures B.1, B.2, and B.3 show the results obtained with the Yo'llava method. Specifically, Figure B.1 reports the accuracy scores per object category when comparing original and generated data; Figure B.2 presents the corresponding precision values; and Figure B.3 illustrates the recall performance across categories.
Similarly, Figures B.4, B.5, and B.6 display the same set of metrics for the RAP-MLLM model. In detail, Figure B.4 provides a breakdown of accuracy, while Figures B.5 and B.6 report precision and recall, respectively, comparing the original and generated datasets concept by concept.

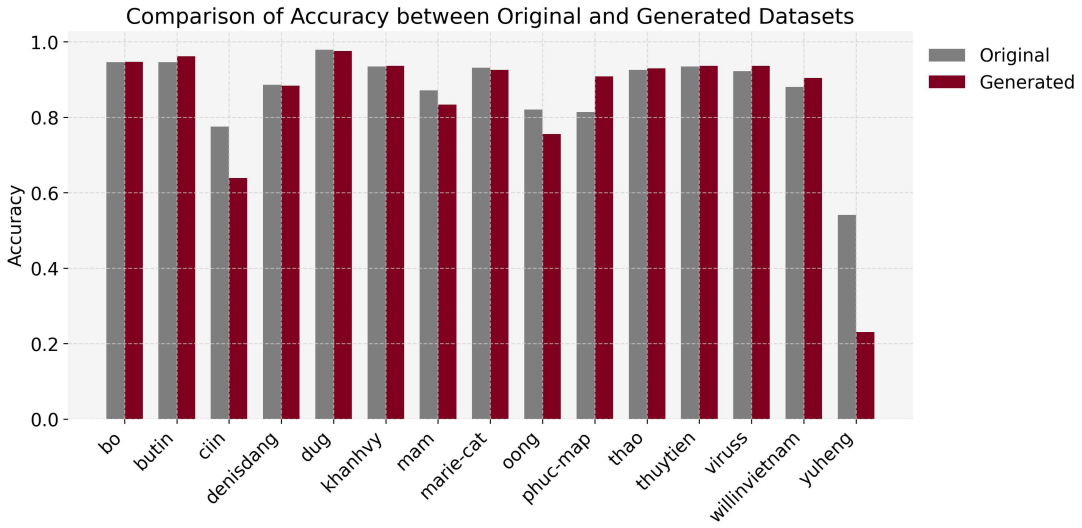


Figure B.1: **Accuracy – Yo'LLaVA.** Comparison of accuracy between the original and the generated dataset, broken down by target concepts. Results are obtained using the Yo'LLaVA method.

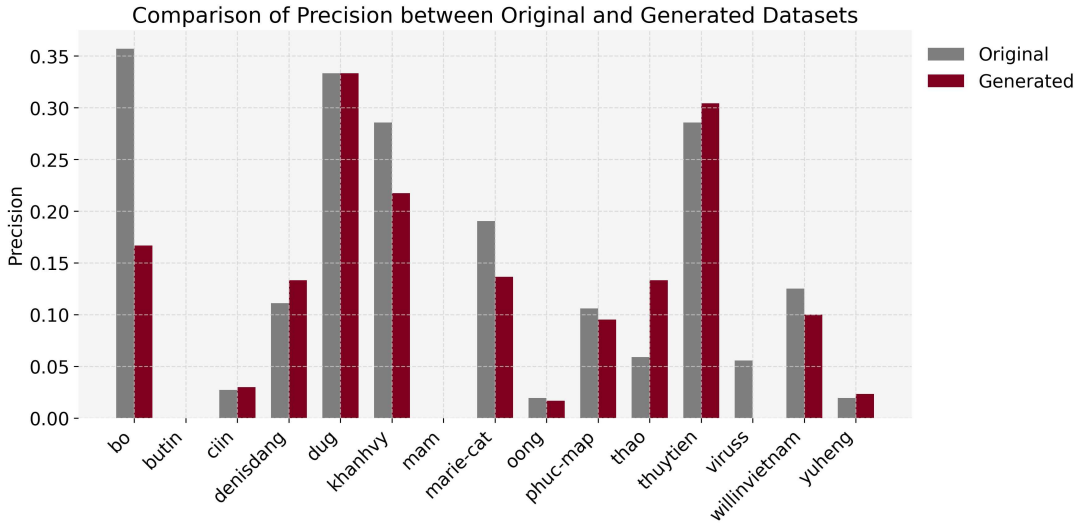


Figure B.2: **Precision – Yo'LLaVA.** Precision analysis for each target concepts, comparing original data with data generated using the Yo'LLaVA method.

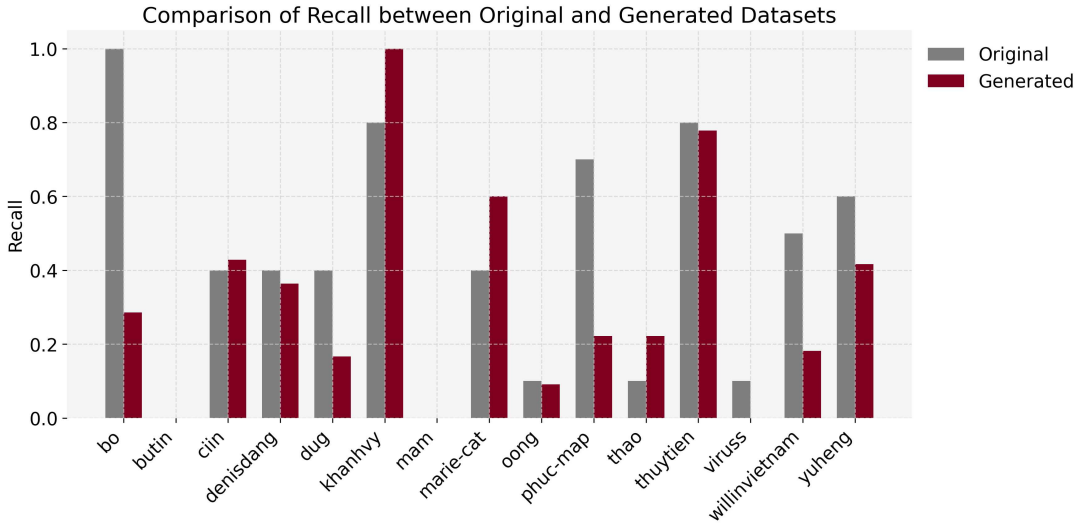


Figure B.3: **Recall – Yo'LLaVA.** Evaluation of recall for each target concepts, comparing the original dataset with the one generated using Yo'LLaVA.

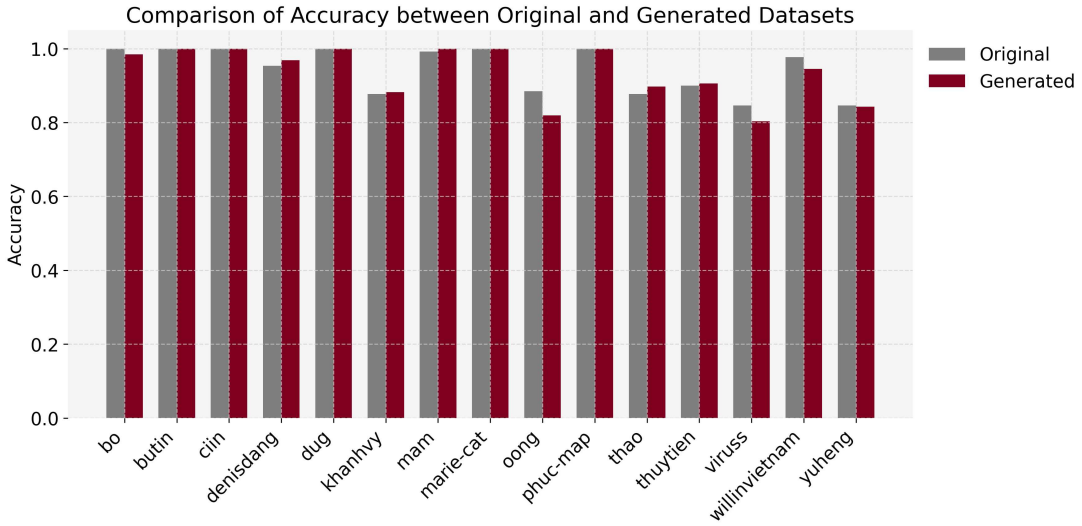


Figure B.4: **Accuracy – RAP-MLLM.** Accuracy comparison between the original and the generated dataset across categories, using the RAP-MLLM method.

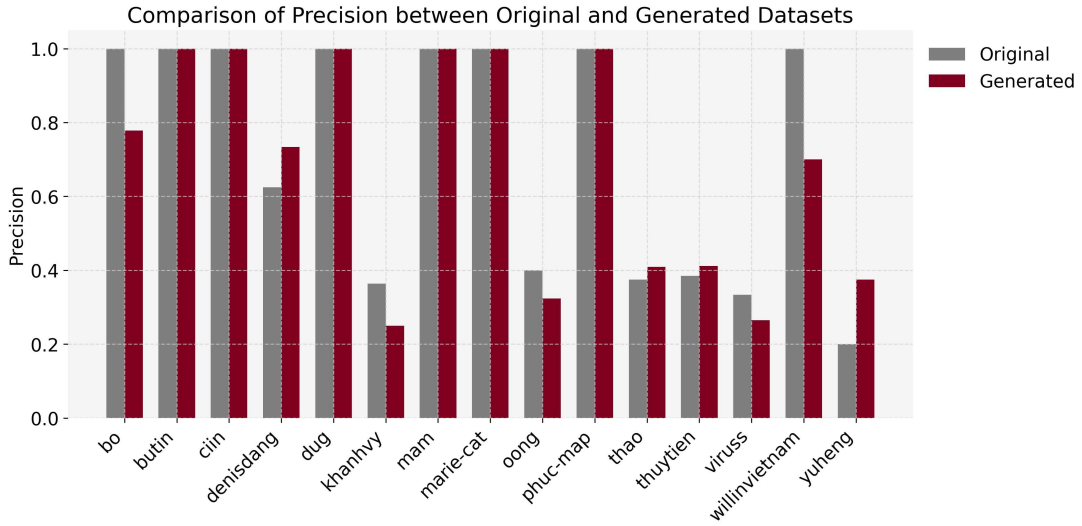


Figure B.5: **Precision – RAP-MLLM.** Comparative precision analysis between original and generated data for each target concepts, using the RAP-MLLM method.

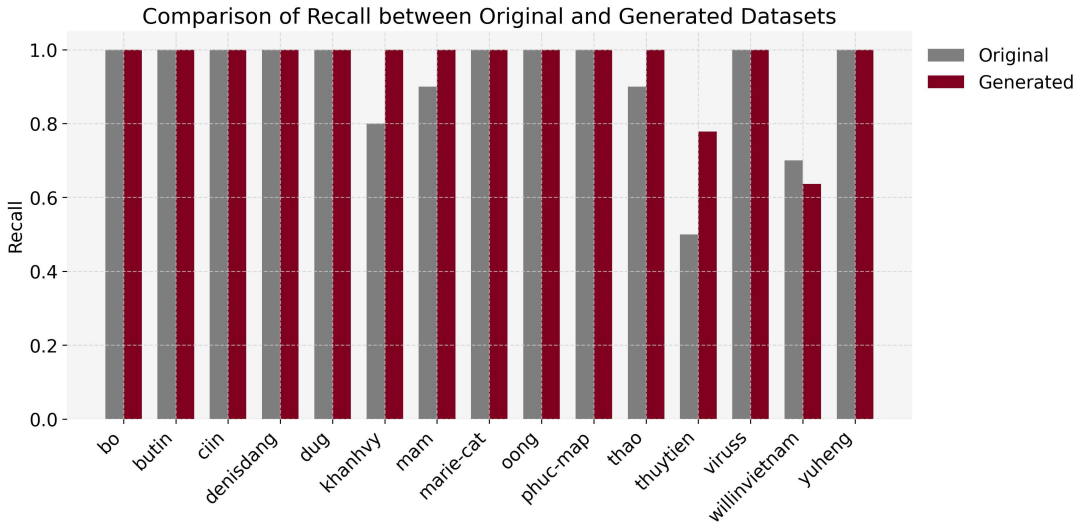


Figure B.6: **Recall – RAP-MLLM.** Recall comparison between the original and generated datasets across categories, obtained using the RAP-MLLM method.

Bibliography

[1] Linda Yilin Wen et al. "Find My Things: Personalized Accessibility through Teachable AI for People who are Blind or Low Vision". In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (May 2024), 403:1–403:6. URL: https://www.microsoft.com/en-us/research/publication/find-my-things-personalized-accessibility-through-teachable-ai-for-people-who-are-blind-or-low-vision/.

[2] Zhang Yu et al. "Personalized Image Semantic Segmentation". In: *ICCV*. 2021.

[3] Renrui Zhang et al. *Personalize Segment Anything Model with One Shot*. May 2023. DOI: 10.48550/arXiv.2305.03048.

[4] L. Tang et al. "Towards Training-Free Open-World Segmentation via Image Prompt Foundation Models". In: *International Journal of Computer Vision* 133.1 (2025), pp. 1–15. DOI: 10.1007/s11263-024-02185-6. URL: https://doi.org/10.1007/s11263-024-02185-6.

[5] Dvir Samuel et al. "Where's Waldo: Diffusion Features For Personalized Segmentation and Retrieval". In: *NeurIPS* (2024).

[6] Niv Cohen et al. ""This Is My Unicorn, Fluffy": Personalizing Frozen Vision-Language Representations". In: *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX*. Tel Aviv, Israel: Springer-Verlag, 2022, pp. 558–577. ISBN: 978-3-031-20043-4. DOI: 10.1007/978-3-031-20044-1_32. URL: https://doi.org/10.1007/978-3-031-20044-1_32.

[7] Y. Alaluf et al. "MyVLM: Personalizing VLMs for User-Specific Queries". In: *Computer Vision – ECCV 2024*. Ed. by A. Leonardis et al. Cham: Springer Nature Switzerland, 2025, pp. 73–91.

[8] Thao Nguyen et al. *Yo'LLaVA: Your Personalized Language and Vision Assistant*. 2024. arXiv: 2406.09400 [cs.CV]. URL: https://arxiv.org/abs/2406.09400.

[9] Soroush Seifi et al. *Personalization Toolkit: Training Free Personalization of Large Vision Language Models*. 2025. arXiv: 2502.02452 [cs.CV]. URL: https://arxiv.org/abs/2502.02452.

[10] Ruichuan An et al. *MC-LLaVA: Multi-Concept Personalized Vision-Language Model*. 2025. arXiv: 2411.11706 [cs.CV]. URL: https://arxiv.org/abs/2411.11706.

[11] Xierui Wang et al. "MS-Diffusion: Multi-subject Zero-shot Image Personalization with Layout Guidance". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=PJqP0wyQek.

[12] Haoran Hao et al. "RAP: Retrieval-Augmented Personalization for Multimodal Large Language Models". In: *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. June 2025, pp. 14538–14548.

[13] Junnan Li et al. "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models". In: *Proceedings of the 40th International Conference on Machine Learning*. ICML'23. Honolulu, Hawaii, USA: JMLR.org, 2023.

[14] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[15] Hyung Won Chung et al. "Scaling instruction-finetuned language models". In: *J. Mach. Learn. Res.* 25.1 (Jan. 2024). ISSN: 1532-4435.

[16] Haotian Liu et al. *Visual Instruction Tuning*. 2023.

[17] Alec Radford et al. "Learning Transferable Visual Models from Natural Language Supervision". In: *International Conference on Machine Learning (ICML)*. 2021.

[18] Wei-Lin Chiang et al. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. https://vicuna.lmsys.org. Accessed: 2023-04-14. 2023.

[19] Christoph Schuhmann et al. "LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models". In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 25278–25294.

[20] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[21] Tianhe Ren et al. *Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks*. 2024. arXiv: 2401.14159 [cs.CV]. URL: https://arxiv.org/abs/2401.14159.

[22] Jianwei Yang et al. *Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V*. 2023. arXiv: 2310.11441 [cs.CV]. URL: https://arxiv.org/abs/2310.11441.

[23] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[24] Scott Reed et al. "Generative Adversarial Text to Image Synthesis". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1060–1069. URL: https://proceedings.mlr.press/v48/reed16.html.

[25] Aditya Ramesh et al. "Zero-Shot Text-to-Image Generation". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 8821–8831. URL: https://proceedings.mlr.press/v139/ramesh21a.html.

[26] Ming Ding et al. "CogView: mastering text-to-image generation via transformers". In: *Proceedings of the 35th International Conference on Neural Information Processing Systems*. NIPS '21. Red Hook, NY, USA: Curran Associates Inc., 2021. ISBN: 9781713845393.

[27] Chenfei Wu et al. "NÜWA: Visual Synthesis Pre-training for Neural visUal World creAtion". In: *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*. Tel Aviv, Israel: Springer-Verlag, 2022, pp. 720–736. ISBN: 978-3-031-19786-4. DOI: 10.1007/978-3-031-19787-1_41. URL: https://doi.org/10.1007/978-3-031-19787-1_41.

[28] Jiahui Yu et al. *Scaling Autoregressive Models for Content-Rich Text-to-Image Generation*. 2022. arXiv: 2206.10789 [cs.CV]. URL: https://arxiv.org/abs/2206.10789.

[29] Robin Rombach et al. "High-Resolution Image Synthesis with Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[30] Aditya Ramesh et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents". In: (Apr. 2022). DOI: 10.48550/arXiv.2204.06125.

[31] Rinon Gal et al. *An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion*. 2022. arXiv: 2208.01618 [cs.CV]. URL: https://arxiv.org/abs/2208.01618.

[32] Nataniel Ruiz et al. "DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 22500–22510. DOI: 10.1109/CVPR52729.2023.02155.

[33] Hu Ye et al. *IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models*. 2023. arXiv: 2308.06721 [cs.CV]. URL: https://arxiv.org/abs/2308.06721.

[34] Nupur Kumari et al. "Multi-Concept Customization of Text-to-Image Diffusion". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 1931–1941. DOI: 10.1109/CVPR52729.2023.00192.

[35] Zeju Qiu et al. *Controlling Text-to-Image Diffusion by Orthogonal Finetuning*. 2024. arXiv: 2306.07280 [cs.CV]. URL: https://arxiv.org/abs/2306.07280.

[36] Weiyang Liu et al. *Parameter-Efficient Orthogonal Finetuning via Butterfly Factorization*. 2024. arXiv: 2311.06243 [cs.LG]. URL: https://arxiv.org/abs/2311.06243.

[37] Ligong Han et al. " SVDiff: Compact Parameter Space for Diffusion Fine-Tuning ". In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2023, pp. 7289–7300. DOI: 10.1109/ICCV51070.2023.00673. URL: https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00673.

[38] Alexander Kirillov et al. "Segment Anything". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023.

[39] Abhishek Dutta and Andrew Zisserman. "The VIA Annotation Software for Images, Audio and Video". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: ACM, 2019. ISBN: 978-1-4503-6889-6/19/10. DOI: 10.1145/3343031.3350535. URL: https://doi.org/10.1145/3343031.3350535.

[40] Brian Chao. *Anime Face Dataset: a collection of high-quality anime faces*. Sept. 16, 2019. URL: https://github.com/bchao1/Anime-Face-Dataset.

[41] Kimmo Karkkainen and Jungseock Joo. "FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 1548–1558.