# A vision-language model as embodied manufacturing agent



## Junyi Qian

Department of Engineering University of Cambridge

This dissertation is submitted for the degree of Master of Philosophy in Machine Learning and Machine Intelligence

Churchill College

August 2024

I would like to dedicate this thesis to my loving parents and grandparents.

### Declaration

I, Junyi Qian of Churchill College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose. This dissertation contains 14,898 words excluding declarations, bibliography, photographs and diagrams, but including tables, footnotes, figure captions and appendices

The code <sup>1</sup> for this project has been developed in Python. Mostly using PyTorch <sup>2</sup>, transformers <sup>3</sup> and datasets <sup>4</sup> libraries. The code implementation of the dissertation is mainly inherented from LLaVA: Large Language and Vision Assistant (H. Liu, Li, Wu, & Lee, 2023) and Masked Autoencoders (He et al., 2021). Extensive modifications have been made to the code bases so as to adapt to the objectives of this dissertation.

Junyi Qian August 2024

<sup>&</sup>lt;sup>1</sup>https://github.com/cam-cambridge/embodied-3d-printer.git

<sup>&</sup>lt;sup>2</sup>https://pytorch.org/

<sup>&</sup>lt;sup>3</sup>https://huggingface.co/docs/transformers/index

<sup>&</sup>lt;sup>4</sup>https://huggingface.co/docs/datasets/index

### Acknowledgements

I would like to extend my heartfelt gratitude to my supervisors, Professor Sebastian Pattinson and Christos Margadji, for their invaluable guidance throughout my research journey. I am deeply thankful to Professor Sebastian Pattinson for his insightful advice and firm support, which provided me with the stability and confidence to overcome the challenges of my research. His meticulous attention to detail and dedication to academic excellence have significantly shaped the direction and quality of my work. I am equally appreciative of Christos Margadji, whose guidance helped me navigate obstacles and offered crucial support during difficult moments. His constant encouragement and thoughtful feedback were instrumental in refining my ideas and keeping me motivated throughout the project. I also wish to express my gratitude to Dr. John Dudley, our course supervisor, for his unwavering support throughout the entire degree and his generosity in providing the resources necessary to complete my dissertation with high quality.

My sincere thanks also go to my fellow students at Churchill College and in the Machine Learning and Machine Intelligence program, including but not limited to Violet Ai, Felix Zhu, Fengzhe Zhang, Curtis Chen, Ruikang Ouyang, and Pablo Monteagudo Lago. Their camaraderie, encouragement, and support have been invaluable throughout this journey.

## Abstract

Additive manufacturing (AM) traditionally relies on predefined parameters and post-production inspections to ensure print quality. This approach, while effective, can be time-consuming, prone to errors, and lacks real-time adaptability to dynamic conditions during the printing process.

This dissertation introduces **LLaVA-IntelliPrint**, the first framework to integrate Vision-Language Models (VLMs) into AM. By combining advanced vision transformers (ViT) and large language models (LLMs), LLaVA-IntelliPrint provides continuous, automated quality assurance during the printing process. The system diagnoses defects and offers natural language corrective suggestions, significantly improving the accuracy, efficiency, and reliability of AM processes.

LLaVA-IntelliPrint leverages self-supervised learning from Masked Autoencoders (MAEs) and cross-modality from LLaVA to estimate process parameters like flow rate, achieving a mean absolute error of **6.49%**, improving on the state-of-the-art (SoTA) of **8.64%**. When integrated with an LLM, it maintains a competitive error rate of **12.02%**, while also offering broader capabilities like extrusion quality classification, error detection, and enhanced human-computer interaction.

Additionally, this thesis presents a **novel architecture** that enhances LLMs' regression capabilities, with potential applications beyond AM. LLaVA-IntelliPrint represents the first application of VLMs in AM, setting a new benchmark for real-time quality assurance in intelligent manufacturing systems.

## **Table of contents**

Li	List of figures xv				
Li	st of t	ables	2	xvii	
No	omen	clature		xix	
1	Intr	oductio	n	1	
	1.1	Additiv	ve Manufacturing	1	
	1.2	Visual-	Language Models (VLMs)	2	
	1.3	Resear	ch Objective	2	
	1.4	Contri	butions	4	
	1.5	Roadm	nap	5	
2	Bac	kground	1	7	
	2.1	Artific	ial Intelligence in Additive Manufacturing	7	
		2.1.1	AI for Quality Assurance	8	
	2.2	Vision	Transformer (ViT)	10	
		2.2.1	Vision Transformer Architecture	10	
		2.2.2	ViT Pretrains	12	
	2.3	Large	Language Models (LLM)	14	
		2.3.1	Architecture of Decoder-Only LLMs	15	
	2.4	Visual	Language Model (VLM)	17	
		2.4.1	Visual Question Answering (VQA)	17	
		2.4.2	Visual Language Model (VLM)	19	
	2.5	Visual-	Language Model as a Quantity Estimator	25	
		2.5.1	Visual Language Model as a Quantity Estimator	25	
		2.5.2	Vision Encoder as a Quantity Regressor	26	
		2.5.3	Large Language Model as Regressor	27	

	2.6	Chapte	er 2 Summary	28
3	Met	hodolog	gy	29
	3.1	Large	Language and Vision Assistant (LLaVA)	29
		3.1.1	Architecture of LLaVA	29
		3.1.2	Training and Performance	30
	3.2	LLaVA	A-IntelliPrint	31
		3.2.1	Four-Stage Training Approach	32
		3.2.2	LLM Regression Head	34
	3.3	3D Pri	inting Dataset	36
		3.3.1	Dataset Introduction	36
		3.3.2	Dataset Statistics	37
	3.4	Trainii	ng Dataset	39
		3.4.1	Conversation Format Modification	39
		3.4.2	Image Pre-Processing	42
	3.5	Maske	ed Autoencoder (MAE)	44
		3.5.1	Pretraining Phase	45
		3.5.2	Fine-tuning Phase	46
	3.6	Evalua	ation Metrics	48
		3.6.1	Masked Autoencoder Performance Evaluation	48
		3.6.2	Visual Language Model Performance Evaluation	49
	3.7	Chapte	er 3 Summary	51
4	Exp	eriment	ts & Results	53
-	<b>4</b> .1	Experi	iment Setup	53
		4.1.1	Train Test Split	53
		4.1.2	MAE Vision Encoder Experiments Overview	54
		4.1.3	LLaVA-IntelliPrint VLM Experiments Overview	54
	4.2	MAE	Vision Encoder Performance	56
		4.2.1	Best Performing Model	56
		4.2.2	Ablation Studies on MAE Architecture	56
		4.2.3	Ablation Studies on Image Augmentation Methods	59
	4.3	LLaVA	A-IntelliPrint VLM Performance	61
		4.3.1	Best Performing Model	61
		4.3.2	LLaVA-IntelliPrint Ablation Study: Experiment 0	62
		4.3.3	LLaVA-IntelliPrint Ablation Study: Experiment 1-6	63
			- I	

		4.3.4	LLaVA-IntelliPrint Ablation Study: Experiment 7-8 Numeric To-	
			kenisation	68
		4.3.5	LLaVA-IntelliPrint Ablation Study: Experiment 9-10 Regression Head	69
		4.3.6	Data Efficiency Optimisation	70
4	1.4	Chapte	er 4 Summary	73
5 (	Con	clusions	s and Future Work	75
5	5.1	Future	Work	76
		5.1.1	Upgrade the Language Model	76
		5.1.2	Post-processing Vision Encoder and Language Model for Enhanced	
			Flow Rate Estimation	76
		5.1.3	Rigorous Testing of the LLM Regression Head	77
		5.1.4	Refining Extrusion Quality Labels	77
		5.1.5	Expanding the Dataset with a Greater Variety of Information	77
		5.1.6	Evaluating the Model on Different Datasets	77
Refe	eren	ces		79

### xiii

# List of figures

1.1	An example of the current state-of-the-art model's performance on an FDM	
	nozzle image	3
1.2	LLaVA-IntelliPrint system flowchart.	4
2.1	Additive Manufacturing Categories	8
2.2	Vision Transformer Architecture (Dosovitskiy et al., 2021)	12
2.3	Decoder-Only LLM Architecture (Touvron et al., 2023)	16
2.4	Examples from the VQA v2.0 dataset (Goyal et al., 2017)	18
2.5	Four architectures for end-to-end VLP	20
2.6	Unified transformer architecture	20
2.7	Fusion-encoder architecture	20
2.8	Dual-encoder architecture	21
2.9	Encoder-decoder architecture	22
2.10	Comparison of VLM on 12 benchmarks (H. Liu, Li, Li, & Lee, 2023)	24
3.1	LLaVA Pretrain Stage (H. Liu, Li, Wu, & Lee, 2023)	31
3.2	LLaVA Finetune Stage (H. Liu, Li, Wu, & Lee, 2023)	31
3.3	LLaVA-IntelliPrint four-stage training	32
3.4	LLaVA-IntelliPrint Architecture	35
3.5	Examples of the Dataset	37
3.6	Dataset dominant colours distribution	38
3.7	Flow Rate Data Distributions	39
3.8	Original Image	43
3.9	Image after preprocessing	43
3.10	Masked Autoencoder Architecture (He et al., 2021)	44
3.11	MAE Fine-tune Architecture	48

4.2	MAE ViT-L reconstruction of 3D Printing Data set based on weights pre-	
	trained on the 3D printing dataset.	57
4.3	MAE ViT-L reconstruction of 3D Printing Data set based on weights pre-	
	trained on ImageNet.	58
4.4	Example of LLaVA-IntelliPrint	61
4.5	Example 1 Image	68
4.6	Example 2 Image	68
4.7	Example 3 Image	68
4.8	Experiment 1: Number of Data Points vs Evaluation Metrics	71
4.9	Experiment 2: Number of Data Points vs Evaluation Metrics	71
4.10	Experiment 3: Number of Data Points vs Evaluation Metrics	72

## List of tables

2.1	Comparisons of self-supervised learning ViT pre-trains on ImageNet-1K	
	Top-1 Accuracy. Where [1] stands for ImageNet 1K; [2] Includes CIFAR-10,	
	CIFAR-100 and STL-10	14
2.2	TOP 10 model accuracy $Acc(ans) = min \left\{ \frac{\#humans that said ans}{3}, 1 \right\}$ in the VQA	
	Challenge 2021 in August 2024 (start from 2021, no end date) (Yadav et al.,	
	2019)	26
3.1	Extrusion Quality Distribution	38
3.2	Flow Rate Data Statistics	39
3.3	Different types of ViT implementations in MAE	45
4.1	Dataset Distribution for Experiments	53
4.2	Key Training Parameters for MAE Pretrain	54
4.3	Key Training Parameters for MAE Finetune	54
4.4	Key Pretraining Parameters for LLaVA-IntelliPrint	55
4.5	Key Finetuning Parameters for LLaVA-IntelliPrint	55
4.6	LLaVA-IntelliPrint Ablation Study	55
4.7	Training Setup A - MAE flow rate regression results: Pretraining on 3D	
	Printing Data; Finetuning on the 3D Printing pre-trained weights	57
4.8	Training Setup B - MAE flow rate regression results: Finetuned on publicly	
	available pretrained weights from ImageNet	58
4.9	Comparison of Regression MAE (Mean ± Std) for Different Augmentation	
	Techniques	60
4.10	LLaVA-IntelliPrint Baseline Setup	62
4.11	LLaVA on 3D Printing Dataset Baseline	63
4.12	LLaVA-IntelliPrint Ablation Study on different training stages	63
4.13	Ablation study results for LLaVA-IntelliPrint, where N/A indicates the mod-	
	els failed to predict any numerical value for the flow rate	64

4.14	LLaVA-IntelliPrint Numeric Tokenisation Ablation Study	69
4.15	Tokenisation Experiment Results	69
4.16	LLaVA-IntelliPrint Regression Head Ablation Study	70
4.17	Regression Head Experiment Results	70
4.18	Standard deviation of evaluation metrics across different experiments	72

## Nomenclature

#### **Acronyms / Abbreviations**

- AI Artificial intelligence
- AM Additive Manufacturing
- CNN Convolutional Neural Networks
- CV Computer Vision
- CV Computer Vision
- FDM Fused Deposition Modeling
- FLP Digital Light Processing
- LLM Large Language Model
- MAE Masked Autoencoders
- MAE Mean Absolute Error
- MLP Multi-Layer Perceptron
- MSE Mean Squarted Error
- NLP Natural Language Processing
- SL Stereolithography
- SoTA State-of-the-art

SSL	Self-supervised Learning
ViT	Vision Transformer
VLM	Visual Language Mode
VLP	Visual Langauge Pretrain

VQA Visual Question Answering

## **Chapter 1**

## Introduction

## 1.1 Additive Manufacturing

Additive Manufacturing (AM), commonly known as 3D printing, is renowned for its capability to manufacture intricate and customised products layer by layer from digital models. This process significantly reduces the time and cost associated with product development, minimises human interaction, and shortens the overall product development cycle. AM might revolutionise some key industries such as aerospace and medical devices (Wong & Hernandez, 2012).

Several types of 3D printers are prevalent, including Fused Deposition Modeling (FDM), Digital Light Processing (DLP), and Stereolithography (SL). FDM, in particular, operates like a hot glue gun with a small nozzle attached to a computer-controlled mechanical carriage that methodically outlines and builds an object layer by layer (Torta & Torta, 2019). The leftmost image in Figure 1.2 depicts a typical FDM printer. This dissertation focuses on FDM printers due to their widespread use, low cost, ease of operation, and compatibility with various materials such as polymers, metals, and ceramics.

However, the inherent complexities of AM processes often lead to high error rates and operational inefficiencies that can undermine its benefits. Currently, the identification and correction of these errors rely heavily on skilled technicians who monitor the AM processes and make adjustments manually. This dependence not only increases production costs due to the need for continuous human oversight but also limits the scalability and speed of AM operations. Furthermore, training operators to achieve proficiency with 3D printing

technologies is a time-consuming and costly endeavour, exacerbated by the rapid evolution of these technologies, which continuously reshapes the required skill sets.

### **1.2** Visual-Language Models (VLMs)

Addressing these challenges, there has been growing interest in developing intelligent error detection and correction systems. Traditional methods have employed various sensors to monitor processes and detect anomalies such as nozzle clogs, including acoustic (Wu et al., 2015), inertial (J. Guo et al., 2019), pressure (Estelle & Gozen, 2024), and current(Wu et al., 2015). However, data from these sensors are often insufficient for comprehensive error detection and correction (Gao et al., 2015).

Recent advancements in artificial intelligence, particularly in natural language processing and computer vision, have paved the way for more sophisticated solutions. Models such as T5 (Raffel et al., 2023), GPT-3 (Brown et al., 2020), and LLAMA (Touvron et al., 2023) have demonstrated the potential of Large Language Models (LLMs) as general-purpose assistants. The introduction of Vision Transformer (ViT) (Dosovitskiy et al., 2021) has shown state-of-the-art capabilities in image feature extractions. Cross-modality mixing techniques, including Flamingo (Alayrac et al., 2022), Contrastive Language-Image Pretraining (CLIP) (Radford et al., 2021), Bootstrapping Language-Image Pre-training (BLIP) (J. Li et al., 2022), and Large Language and Vision Assistant (LLaVA) (H. Liu, Li, Wu, & Lee, 2023), show great potential for visual-language interaction.

Current deep learning models mainly utilise convolutional neural networks (CNNs) for realtime defect detection in additive manufacturing, addressing issues such as nozzle clogs and layer misalignment (Brion & Pattinson, 2022b; Brion et al., 2022; Farhan Khan et al., 2021; Margadji et al., 2024). However, CNNs pose challenges in scalability, and the performance is not always satisfying. Vision Transformers (ViTs), which utilise self-attention mechanisms, offer promising alternatives but remain largely unexplored in this context. Additionally, integrating language models for human-computer interaction can enhance real-time system usability by providing explanations and suggestions, thus bridging automated detection and human expertise. This potential remains unexplored.

### **1.3 Research Objective**

The objective of this thesis is to develop a Vision-Language Model (VLM) that comprehends the Fused Deposition Modelling (FDM) printing process, estimates key parameters, associates

visual features with textual outputs, and suggests specific actions. Current state-of-the-art VLMs, such as GPT-40 and LLaVA 1.5, cannot interpret nozzle images or estimate flow rates, as shown in Figure 1.1. This highlights a clear application and research gap in the field.



**Prompt:** Estimate the flowrate and Describe the following image in detail

- ChatGPT-40: The image shows a close-up of a 3D printing process. You can see the nozzle of the 3D printer extruding a blue filament, creating distinct layer lines typical of FDM printing. To estimate the flow rate, you'd need details like nozzle diameter, print speed, layer height, and extrusion width. The flow rate is typically calculated by multiplying layer height, extrusion width, and print speed. Without these specific values, a precise flow rate can't be determined from the image alone.
- LLaVA: I'm sorry, but I cannot estimate the flowrate or describe the image as it is not provided in the question. Please provide more information or context so I can assist you better.
- LLaVA-IntelliPrint (Our Model): The focal point of this image is the extrusion nozzle, which suggests the operation of a material extrusion process, possibly within additive manufacturing. There are clear indications of over-extrusion from this extrusion nozzle, potentially operating at 238 % flowrate. Over-extrusion can occur due to a high feedrate or excessive temperature, leading to problems like nozzle clogging or inadequate surface finish.
- **Reference:** Let's examine the nozzle of this FDM 3D printer. There are indications of **over-extrusion** from this extrusion nozzle, possibly operating at 235 % flowrate. Over-extrusion occurs when the feedrate is too high or the temperature is excessive, causing excessive material deposition and potential issues like nozzle clogging or poor surface finish.

Fig. 1.1 An example of the current state-of-the-art model's performance on an FDM nozzle image.

This dissertation introduces **LLaVA-IntelliPrint**, a novel integration of advanced visionlanguage models (VLMs) and large language models (LLMs) into 3D printers. By equipping these printers with enhanced vision and language processing capabilities, they can autonomously monitor, diagnose, and correct errors in real time, thereby improving the efficiency and reliability of additive manufacturing.

As illustrated in Figure 1.2, the system utilizes Vision-Language Models (VLMs) to:

- 1. Describe overall printing quality (e.g., over-extrusion, optimal extrusion, under-extrusion).
- 2. Estimate the current printing flow rate.
- 3. Detect errors and analyse their causes using natural language processing.
- 4. Provide corrective suggestions for identified errors.



**Q: What is the G-code to correct it?** A: M221 S110

O: Describe the manufacturing process.

Fig. 1.2 LLaVA-IntelliPrint system flowchart.

This work demonstrates the power of integrating VLMs with language models to create a more intelligent, user-friendly, and efficient 3D printing system. By combining visual analysis with natural language processing, LLaVA-IntelliPrint autonomously identifies and explains errors, offers actionable suggestions, and provides operators with valuable insights. This approach not only reduces the learning curve and training costs but also enhances decision-making, productivity, and the overall reliability of additive manufacturing processes.

## 1.4 Contributions

The main contributions of this dissertation include:

- State-of-the-art Vision Encoder for flow rate estimation: Developed a transformerbased self-supervised learning method (Masked Autoencoder) to estimate printing flow rate with a Mean Absolute Error of 6.49%, surpassing previous CNN-based approaches (Margadji et al., 2024).
- First VLM in Additive Manufacturing: Introduced VLM in the field of Additive Manufacturing, significantly improving error detection accuracy and effectiveness compared with existing methods and offering functionalities such as interpretability and human-computer interaction that were previously unavailable.
- VLM for Regression: Introduced a novel architecture that enables VLMs to perform quantitative regression, allowing the estimation of numerical properties from images (e.g., determining flow rate from a nozzle image).

### 1.5 Roadmap

This thesis is organised into five chapters, each addressing a critical component of the research process, from background knowledge to the development and evaluation of the proposed system.

- **Chapter 1: Introduction** Introduces additive manufacturing and Vision-Language Models (VLMs), outlining the research objectives and contributions.
- Chapter 2: Background Reviews related literature on AI techniques in manufacturing, Vision Transformers (ViTs), and Large Language Models (LLMs), focusing on the integration of these technologies into VLMs.
- Chapter 3: Methodology Details the design and implementation of the LLaVA-IntelliPrint system, including the architecture, dataset, training process, and evaluation metrics.
- Chapter 4: Experiments and Results Presents the experimental setup, results, and analysis, including performance evaluation and ablation studies of the LLaVA-IntelliPrint system.
- Chapter 5: Conclusion and Future Work Summarises the research findings, discusses contributions, and proposes directions for future research.

## Chapter 2

## Background

This chapter reviews the relevant literature, including existing artificial intelligence techniques in additive manufacturing, the architecture of Vision Transformers (ViTs) and Large Language Models (LLMs), and the integration of these technologies into VLMs. The chapter also discusses the challenges and opportunities in using VLMs for regression tasks within the context of additive manufacturing.

## 2.1 Artificial Intelligence in Additive Manufacturing

Additive manufacturing (AM) technology has seen significant advancements over the past decade, finding applications in fields such as medicine, aerospace, and construction (Wong & Hernandez, 2012). As the technology's use expands, its maturity continues to improve. Ensuring that AM products meet mechanical, material, and functional design standards requires a thorough understanding of the technology's characteristics, careful selection of suitable materials and processes, and optimisation of process parameters. To this end, various methodologies have been developed to enhance the effective application of AM technology.

One particularly impactful development is the integration of artificial intelligence (AI) through machine learning, which enables the detailed analysis of the physical phenomena associated with AM. Gu et al. (2023) categories AI techniques in AM into four main areas Figure 2.1.

**The first area** involves AI for design optimisation specific to AM technology, streamlining the design process. (Goh et al., 2021; J. Jiang et al., 2022).



Fig. 2.1 Additive Manufacturing Categories

**The second area** uses AI to develop and analyse new materials for AM, focusing on material properties. Various AI techniques have been proposed for material selection, development, and properties prediction based on simulations of existing materials synthesis (C.-T. Chen & Gu, 2019).

**The third area** includes AI for quality assurance and real-time optimisation. By applying AI techniques, models on the relationship between process variables and output quality are generated. This approach allows for various output quality indicators such as dimensions (C.-T. Chen & Gu, 2019), surface quality (Aoyagi et al., 2019), mechanical properties (Sood et al., 2012), density (Mbodj et al., 2021), and microstructure (Suzuki et al., 2022) to be applied.

**The fourth area** employs AI for predicting output quality, aiming to ensure the quality certification of finished AM products (Equbal et al., 2011; Shen et al., 2004; R.-J. Wang et al., 2009).

### 2.1.1 AI for Quality Assurance

#### **Parameters to Estimate**

The focus of this dissertation is on ensuring real-time AM printing quality, falling under the third category of AI applications in AM. The key objective for 3D printing quality assurance is estimating the printing parameters such as the actual and target temperatures for the hot-end and bed, flow rate, lateral speed and Z offset. (Brion & Pattinson, 2022b).

Out of the printing parameters, the flow rate, or the extrusion multiplier, determines the amount of plastic to extrude (Wenger et al., 2022). The correct flow rate is crucial for dimensionally accurate 3D prints. A flow rate greater than, approximately equal to, and less than 100% is classified as over-extrusion, optimal-extrusion, and under-extrusion, respectively. If

the flow rate is too high, over-extrusion occurs, causing issues like excessive filament use, blobbing, stringing, and poor dimensional accuracy. Conversely, a low flow rate leads to under-extrusion, resulting in problems like gaps, weak layer adhesion, and poor surface finish (Torta & Torta, 2019). Compared to controlling temperature, which involves heating the hot-end and bed, adjusting the flow rate by sending G-code to the stepper motors is faster and easier, and it provides more immediate feedback (Torta & Torta, 2019). This dissertation will focus on estimating and correcting the printing flow rate, while the estimation of temperature will be considered for future work.

#### **3D Printing Quality Assurance Models**

Various indirect methods have been developed to estimate parameters and detect errors during additive manufacturing, such as monitoring acoustic emissions (Wu et al., 2015), printer vibrations (Rao et al., 2015), inertial measurements (J. Guo et al., 2019), pressure (Estelle & Gozen, 2024), and motor current (Wu et al., 2015). However, data from these sensors often lack the richness needed for comprehensive error detection and correction, relying on accurate physical models and costly equipment. Vision sensors, on the other hand, provide more detailed information, enabling the detection of larger defects like layer shifts and low-quality infills using traditional computer vision techniques (He et al., 2019; Huang et al., 2021; Straub, 2015). Multi-camera systems, which offer perspectives beyond a single visible-spectrum camera, have also been explored (Cunha et al., 2021; Holzmond & Li, 2017; Straub, 2015), allowing for more detailed error detection. However, these approaches are often expensive, require complex calibration, are sensitive to lighting and surface properties, and may be limited by scanner resolution.

Recently, deep learning-based vision approaches have been used for autonomous error correction after printing, effectively addressing errors that develop as the material cools, such as cracking and warping (Jin et al., 2020; Z. Zhang et al., 2020). To further reduce and correct errors during printing, vision technology can be integrated with traditional real-time feedback loop strategies. Brion et al. (2022) adapted Convolutional Neural Networks (CNNs) and multi-head neural networks for this purpose, also creating a specialised image dataset for extrusion quality classification. These models can estimate material flow rate categories (e.g., over-extrusion, good extrusion, under-extrusion) during printing. Using the self-regulated network 8GF (RegNet) (Xu et al., 2021) backbone, their model achieved a mean squared error of 92 and a mean absolute error of 9.5 % in flow rate prediction (Brion & Pattinson, 2022a).

Margadji et al. (2024) proposed an iterative learning framework that improves performance by learning from its own errors during repeated build cycles of the same part. This iterative learning approach has been shown to enable robust error detection and correction, reaching a mean absolute error of 8.64 % in estimating the extrusion flow rate. This performance was achieved by using a RegNet as the convolutional backbone, which was then fine-tuned on 15 typical 3D printing geometries.

With the introduction of the attention mechanism (Vaswani et al., 2023), Vision Transformers (ViTs) have been used as powerful encoders to capture image visual information (Dosovitskiy et al., 2021). However, a Vision Transformer-based model has not yet been developed to estimate 3D printing parameters (i.e., extrusion flow rate). One objective of this thesis is to implement a ViT-based model on 3D printing nozzle images and evaluate its performance in estimating the flow rate.

### 2.2 Vision Transformer (ViT)

Vision Transformers (ViTs) adopt an attention-based architecture that has rapidly become a leading deep-learning model for computer vision tasks. Notably, Cordonnier et al. (2020) theoretically demonstrated the equivalence between multi-head self-attention mechanisms and Convolutional Neural Networks (CNNs), laying the groundwork for subsequent developments. Building on this foundation, Dosovitskiy et al. (2021) extended the Transformer architecture (Vaswani et al., 2023) to facilitate large-scale pertaining.

Along with derived pretraining methods such as CLIP (Radford et al., 2021), MAE (He et al., 2021), and BEiT (Bao et al., 2022), as well as advancements in language-augmented foundational vision models (Gan et al., 2022; C. Li et al., 2022), ViT has significantly enhanced capabilities in visual understanding tasks. These tasks encompass classification (Pham et al., 2023; Radford et al., 2021; J. Yang et al., 2022; Yuan, Chen, et al., 2021), detection (L. H. Li et al., 2022; Y. Zhong et al., 2021), segmentation (C. Li, Gan, et al., 2023; H. Zhang et al., 2023), and captioning (J. Li et al., 2023; J. Wang et al., 2022), alongside visual generation and editing (Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022).

### 2.2.1 Vision Transformer Architecture

Figure 2.2 illustrates the ViT architecture, highlighting its key components:

 Patch Partitioning: Given an image x ∈ ℝ<sup>H×W×C</sup>, it's divided into a grid of nonoverlapping patches, each of size P×P. The number of patches N is:

$$N = \frac{H \times W}{P^2} \tag{2.1}$$

where *H* and *W* are the image's height and width, *P* is the patch size, and *N* denotes the total number of patches. Each patch is embedded into a vector of dimension *D*, forming a sequence of embeddings  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N] \in \mathbb{R}^{N \times D}$ .

Position Embedding: Positional information is added by a learnable position embedding P ∈ ℝ<sup>N×D</sup>:

$$\mathbf{Z}^{\text{pos}} = \mathbf{Z} + \mathbf{P} \tag{2.2}$$

where  $\mathbf{Z}^{\text{pos}}$  represents the position-enhanced patch embeddings.

• [CLS] Token: A learnable classification token  $\mathbf{z}_{cls} \in \mathbb{R}^D$  is prepended to the sequence:

$$\mathbf{Z}^{0} = [\mathbf{z}_{\text{cls}}, \mathbf{z}_{1}^{\text{pos}}, \mathbf{z}_{2}^{\text{pos}}, \dots, \mathbf{z}_{N}^{\text{pos}}]$$
(2.3)

where  $\mathbf{Z}^0$  serves as the input to the Transformer encoder.

- **Transformer Encoder:** The encoder, consisting of multiple layers, includes selfattention and feedforward networks.
  - Self-Attention: The input sequence  $\mathbf{Z} \in \mathbb{R}^{(N+1) \times D}$  is projected to query, key, and value matrices:

$$\mathbf{Q} = \mathbf{Z}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{Z}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{Z}\mathbf{W}_V \tag{2.4}$$

where  $\mathbf{W}_Q$ ,  $\mathbf{W}_K$ , and  $\mathbf{W}_V$  are learnable projection matrices. Attention scores are computed as:

Attention(
$$\mathbf{Q}, \mathbf{K}, \mathbf{V}$$
) = softmax  $\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}$  (2.5)

with  $d_k$  being the dimension of the key vectors.

 Feedforward Neural Network: The output of the self-attention is passed through a feedforward network:

$$FFN(\mathbf{z}) = ReLU(\mathbf{z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$
(2.6)

where  $W_1$  and  $W_2$  are weight matrices, and  $b_1$  and  $b_2$  are bias vectors.

• **Classification Head:** The [CLS] token from the final layer *L* is passed through an MLP head for classification:

$$\mathbf{y} = \operatorname{softmax}(\mathbf{z}_{cls}^{L}\mathbf{W}_{cls} + \mathbf{b}_{cls})$$
(2.7)

where  $\mathbf{W}_{cls} \in \mathbb{R}^{D \times K}$  and  $\mathbf{b}_{cls} \in \mathbb{R}^{K}$  are the weights and bias of the classification head, *K* is the number of classes, and **y** represents the predicted class probabilities.



Fig. 2.2 Vision Transformer Architecture (Dosovitskiy et al., 2021)

### 2.2.2 ViT Pretrains

Due to the intensive computational resources required in training Vision Transformers (ViTs), which typically involve millions of parameters, pretraining ViTs has become a common practice for feature representation that can be used in downstream tasks. This involves training them on large datasets to learn rich, general visual representations. The pretraining phase enables the models to develop a strong understanding of visual features and spatial hierarchies. Transfer learning then fine-tunes these pre-trained models on specific tasks, leveraging the general knowledge gained during pretraining to achieve high performance with relatively less data and computational effort. Some common ViT pretraining approaches include:

#### **Supervised Pre-Training**

• ViT (Vision Transformer): Supervised training directly on the [CLS] token of ViT on large labelled datasets like ImageNet.

#### **Self-Supervised Pre-Training**

Self-supervised learning (SSL) involves training a model to predict part of the input from other parts, which does not require labelled data. Common self-supervised learning techniques include Context as supervision (i.e., predicting the relative location between patches), Inpainting (i.e., filling in masked parts), and Colourisation (i.e., adding colour to a grayscale image) (Goodfellow et al., 2017).

- MAE (Masked Autoencoders) (He et al., 2021): Trains the model to reconstruct masked parts of the input image. The masking ratio is usually 75%.
- **BEiT (Bidirectional Encoder Representation from Image Transformers)** (Bao et al., 2022): Similar to MAE, BEiT masks random patches of the input image. However, instead of reconstructing image pixels, it predicts discrete tokens for these masked patches using a discrete variational autoencoder (dVAE) tokenizer.
- **DINO** (Self-Distillation with No Labels) & DINO v2 (Caron et al., 2021) (Oquab et al., 2024): Uses self-distillation with knowledge distillation, where a model is trained to match the output distribution of a teacher model without using labelled data. On top of the original DINO, DINO v2 automates the data curation, which enhances stability and training efficiency.
- Momentum Contrast (MoCo) v3 (X. Chen et al., 2021): Employs both self-supervised learning (SSL) and contrastive learning. It takes two crops of each image under random data augmentation, encodes them with separate encoders, and minimises the contrastive loss between the two outputs.

#### **Contrastive Learning**

Contrastive learning involves training a model to distinguish between similar and dissimilar pairs of data points. It often uses a loss function that encourages the model to bring similar data points closer and push dissimilar ones apart (Goodfellow et al., 2017).

• CLIP (Contrastive Language-Image Pre-Training) (Radford et al., 2021): CLIP learns visual concepts from natural language descriptions by training on a large dataset of images and their corresponding text descriptions.

#### **Other Learning Models**

• Swin Transformer (Shifted Windows Transformer) (Z. Liu et al., 2021): Introduces a hierarchical architecture with shifted windows for local and global self-attention,

improving efficiency and scalability for dense prediction tasks like object detection and segmentation.

• **DeiT (Data-efficient Image Transformers)** (Touvron, Cord, Douze, et al., 2021): Enhances the data efficiency of vision transformers by integrating knowledge distillation from a convolutional neural network teacher, which provides additional supervision to the transformer during training.

Method	pre-train data	ViT-B	ViT-L/16	ViT-H
GPT (M. Chan et al. 2020)	IN1K <sup>1</sup> +	68.7	72.0	72.6
101 1 (W. Chen et al., 2020)	Labelled Data <sup>2</sup>			72.0
DINO (Caron et al., 2021)	IN1K	82.8	-	-
MoCo v3 (X. Chen et al., 2021)	IN1K	83.2	84.1	-
BEiT (Bao et al., 2022)	IN1K+DALLE	83.2	85.2	-
MAE (He et al., 2021)	IN1K	83.6	85.9	86.9

Table 2.1 Comparisons of self-supervised learning ViT pre-trains on ImageNet-1K Top-1 Accuracy. Where [1] stands for ImageNet 1K; [2] Includes CIFAR-10, CIFAR-100 and STL-10

These ViT pre-training methods are strong candidates for use as vision encoders in VLMs. Notably, self-supervised learning approaches stand out, especially given the high cost of acquiring labelled 3D printing data described in natural language. The ability of these methods to learn without labelled data makes them particularly well-suited for the objectives of this dissertation.

The Masked Autoencoder (MAE) is chosen as the vision encoder in this dissertation due to its outstanding performance on the ImageNet-1K dataset (Table 2.1). MAE excels in classification accuracy, simplicity, and stability, making it superior to other self-supervised learning methods. Its masking and reconstruction approach effectively learns rich visual representations, leading to robust generalisation across tasks. The simplicity of MAE's architecture also ensures ease of implementation and reliability. making it an ideal vision encoder for this research. By leveraging MAE as the vision encoder, this research ensures that the model benefits from robust visual features, enhancing performance in tasks like flow rate estimation and error detection in additive manufacturing.

## 2.3 Large Language Models (LLM)

Large Language Models (LLMs), on the other hand, have gained significant attention due to their universal interface and capability as general-purpose assistants based on instruction.

14

These models are based on Transformer architecture (Vaswani et al., 2023), with state-ofthe-art modles including BERT (Devlin et al., 2019), T5 (Raffel et al., 2023), GPT-3 (Brown et al., 2020), Mistral (A. Q. Jiang et al., 2023), Chinchilla (Hoffmann et al., 2022), PaLM (Chowdhery et al., 2022) and LLaMA (Touvron et al., 2023).

LLMs can be classified into two main types:

- **Decoder-Only Models:** These models, such as GPT (Radford et al., 2021), use only the decoder part of the Transformer, generating text in an autoregressive manner by predicting tokens sequentially. They excel at tasks like text generation and completion.
- Encoder-Decoder Models: Models like BERT (Devlin et al., 2019) and T5 (Raffel et al., 2023) employ both the encoder and decoder components. The encoder processes the input sequence to create context-aware representations, while the decoder generates the output sequence. These models are particularly effective for tasks such as translation and summarisation.

### 2.3.1 Architecture of Decoder-Only LLMs

State-of-the-art models such as GPT (Radford et al., 2021) and LLaMA (Touvron et al., 2023) are prominent examples of Large Language Models (LLMs) that have demonstrated exceptional capabilities in natural language processing tasks. These models primarily utilise a decoder-only architecture, which is particularly effective for generating coherent and contextually relevant free-text outputs from a wide range of input prompts. The decoder-only architecture is designed to predict the next token in a sequence, making it highly suitable for tasks that involve text generation, summarisation, translation, and conversational agents. LLaVA-IntelliPrint could leverage the capability of decoder-only LLMs and generate concise descriptions on 3D printing images. As illustrated in Figure 2.3, key steps in processing text data include:

- **Tokenisation :** Text is divided into tokens, representing words, subwords, or characters. Subword tokenisation methods, such as Byte Pair Encoding (BPE) (Goodfellow et al., 2017), are common in LLMs.
- Embedding: Tokens are converted into high-dimensional vectors called embeddings:

$$\mathbf{e}_i = \mathbf{W}_e t_i \tag{2.8}$$

where  $\mathbf{W}_e$  is the embedding matrix and  $t_i$  is the ith language token.



Fig. 2.3 Decoder-Only LLM Architecture (Touvron et al., 2023)

• **Positional Embedding:** Positional information is added to token embeddings to reflect the order of tokens in a sequence:

$$\mathbf{e}_i' = \mathbf{e}_i + \mathbf{p}_i \tag{2.9}$$

Where  $\mathbf{p}_i$  is the positional information.

• **Self-Attention Mechanism:** The Transformer's core innovation, self-attention, assigns importance to different tokens based on their relevance:

Attention(
$$\mathbf{Q}, \mathbf{K}, \mathbf{V}$$
) = softmax  $\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$  (2.10)

where Q, K, and V are the query, key, and value matrices derived from the embeddings, and  $d_k$  represents the dimensionality of the key vectors.
• Feedforward Network: The output is processed through a feedforward neural network:

$$FFN(\mathbf{x}) = (\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$
(2.11)

where  $W_1$  and  $W_2$  are weight matrices, and  $b_1$  and  $b_2$  are bias vectors.

• **Softmax Function:** The final output is converted into a probability distribution over the vocabulary:

$$P(y_i = j | \mathbf{x}) = \frac{\exp(\mathbf{z}_j)}{\sum_{k=1}^{V} \exp(\mathbf{z}_k)}$$
(2.12)

where  $\mathbf{z}_j$  is the logit corresponding to token *j*, and *V* is the size of the vocabulary.

• **Cross-Entropy Loss:** During training, the model's predictions are evaluated using cross-entropy loss:

$$\text{Loss} = -\sum_{i=1}^{V} y_i \log(p_i)$$
(2.13)

where  $y_i$  represents the true distribution (which is 1 for the correct token and 0 for others), and  $p_i$  denotes the predicted probability for the *i*-th token in the vocabulary. Minimising this loss aids the model in improving its predictions by increasing the probability of the correct next token.

A thorough understanding of LLM architecture is crucial for effectively exploring and optimising Vision-Language Models (VLMs). By tailoring and refining LLMs for specific tasks, we can enhance model performance and achieve superior outcomes in targeted applications. In this research, we hypothesise that modifying the tokeniser and loss function in the LLM could significantly improve its numeric regression capabilities. Further details on these modifications will be discussed in Chapter 3.

# 2.4 Visual-Language Model (VLM)

After reviewing state-of-the-art vision encoders and language models, this section aims to present and analyse the common architectures and methods in the literature that combine these modalities to achieve multimodal functionalities.

### 2.4.1 Visual Question Answering (VQA)

With the advancements in Computer Vision (CV) and Natural Language Processing (NLP), tasks requiring the understanding of images described by natural language have become increasingly prevalent. Common applications include image captioning (Anderson et al.,

2018), visual grounding (Fukui et al., 2016), visual question answering (VQA) (Agrawal et al., 2016), and visual dialogue (Das et al., 2017; D. Guo et al., 2019). Image captioning involves describing the content and context of an image. VQA, on the other hand, is the task of answering open-ended questions based on an image, producing natural language responses to natural language questions.

The objective of this dissertation—developing a vision-language model as an embodied manufacturing agent—falls at the intersection of VQA and visual dialogue. This project involves describing printing quality, identifying and correcting errors through natural language, and engaging in multiple rounds of conversation, including queries about flow rate, extrusion quality, and printing corrections.

# **VQA Dataset**

As shown in Figure 2.4, VQA datasets typically include an image, a question, and a corresponding gold-standard answer. Several types of VQA datasets exist, including:



Fig. 2.4 Examples from the VQA v2.0 dataset (Goyal et al., 2017)

- General VQA: Includes datasets such as VQA v2.0 (Goyal et al., 2017), MovieQA (Tapaswi et al., 2016), and Visual Question Answering under Changing Priors (VQA-CP) (Agrawal et al., 2018).
- Scene-specific VQA Datasets: These require spacial understanding, such as CLEVR (Johnson et al., 2016), GQA (Hudson & Manning, 2019), and ShapeWorld (Kuhnle & Copestake, 2017).

- **Knowledge-Based VQA**: These datasets necessitate external knowledge not directly evident from the image, including OK-VQA (Marino et al., 2019), FVQA (P. Wang et al., 2017), and A-OKVQA (Schwenk et al., 2022).
- Other Domain-Specific VQA Datasets: These focus on particular domains, such as TextVQA (Singh et al., 2019), which concentrates on text understanding, and VQA-MED-2021 (Abacha et al., n.d.), which focuses to VQA in the medical field.

# 2.4.2 Visual Language Model (VLM)

To tackle VQA problems, Visual Language Models (VLMs) are employed. A vision-language model is an integration of vision and natural language models. It processes images and their respective textual descriptions as inputs and learns to associate knowledge from both modalities. The vision component captures spatial features from the images, while the language model generates text from encoded information.

Vision-language models (VLMs) combine vision models with language models through various techniques and architectures. Pre-training VLMs is crucial for integrating visual and textual data, enabling these models to perform tasks such as image captioning and visual question answering effectively. This process allows the model to learn shared representations for images and text, enhancing its ability to generate accurate descriptions and understand complex queries. Pre-training on large-scale datasets ensures robust generalisation across various tasks without the need for extensive task-specific data for downstream applications, which require further fine-tuning. Based on training techniques, vision-language pretraining (VLP) can be classified into End-to-End VLP and Modular VLP.

#### **End-to-End Visual-Language Pretraining**

End-to-end vision-language pretraining (VLP) involves training all components simultaneously on the same data and objectives. Depending on how modalities are combined, end-to-end VLP can be classified into four architectures (Figure 2.5).









Fig. 2.6 Unified transformer architecture

Fig. 2.7 Fusion-encoder architecture

- Unified transformer architecture employs transformer-based systems to process and fuse image and text embeddings. As shown in Figure 2.6, this modality is usually combined before feeding into transformers. Notable contributions include the work by Zhou et al. (2019), which first demonstrates improved performance in image captioning and visual question answering through unified vision-language pre-training. W. Wang et al. (2022) proposed BEiT-3, which treats images similarly to text by dividing them into patches and applying a transformer model. This model masks certain tokens, learning to predict them, effectively capturing contextual relationships within images. The BLIP framework by J. Li et al. (2022) introduced an image-grounded text encoder and decoder, achieving state-of-the-art results.
- Fusion-encoder architecture has separate encoders for modalities and integrates them at an intermediate stage by cross-attention before producing the final prediction by a multimodal encoder (Figure 2.7). Notable models include Learning Cross-Modality

Encoder Representations from Transformers (LXMERT) (Tan & Bansal, 2019), which uses cross-attention layers for tasks like visual question answering, while Align Before Fuse (ALBEF) (J. Li et al., 2021) aligns modalities using cross-modal contrastive loss before fusing, enhancing joint representation learning.

• **Dual-Encoder Architecture** leverages distinct encoders for processing visual and textual data independently, with their outputs integrated at a subsequent stage. As illustrated in Figure 2.8, this architecture includes separate encoders for images and text, and employs contrastive learning to effectively distinguish between similar and dissimilar pairs of data points. Prominent examples include Contrastive Language–Image Pre-training (CLIP) (Radford et al., 2021), which employs contrastive learning on a large-scale dataset of images and captions to generate joint visual-textual embeddings. The Large-scale ImaGe and Noisy-text embedding (ALIGN) model (Jia et al., 2021) extends this approach by using an even larger and more uncurated dataset to enhance model robustness. Furthermore, Contrastive Captioning (CoCa) (Yu et al., 2022), which focuses on image captioning, integrates contrastive learning with caption generation, resulting in outputs that are more contextually relevant.



Fig. 2.8 Dual-encoder architecture

• Encoder-decoder architecture involves two stages: the encoder converts input data into a latent representation, and the decoder transforms this representation into the target output. Representative models include Vision-and-Language BERT (ViLBERT) (Lu et al., 2019), which extends BERT to jointly encode visual and textual inputs before decoding for various tasks; Simple Visual Language Model Pretraining with Weak

Supervision (SimVLM) (Z. Wang et al., 2022), which employs weakly supervised data to enhance visual-language pretraining; and Pathways Language and Image model (PaLI) (X. Chen et al., 2023), designed for multilingual and multimodal tasks.



Fig. 2.9 Encoder-decoder architecture

Of the four architectures described above, the **Unified Transformer architecture** and the **Encoder-Decoder architecture** are the most suitable for our project, as both involve decoders for natural language generation. These models leverage decoders to produce free-form text, enabling more flexible and detailed outputs. In contrast, while the Dual-Encoder and Fusion-Encoder architectures can be extended to support text generation, they are primarily designed for tasks involving the matching of images with predefined labels, making them less suitable for our objective of generating free text descriptions of 3D printing processes. In this dissertation, we explore the Unified Transformer architecture due to its inherent capability to handle complex, multimodal data and generate coherent textual descriptions. The Encoder-Decoder model, though promising, is reserved for future work, as it typically requires more extensive training data and computational resources to achieve similar levels of performance in multimodal text generation tasks.

The exploration of these architectures leads us to consider the broader landscape of visionlanguage pre-training (VLP) methods, which commonly perform end-to-end pre-training using large-scale image-text pair datasets. However, as model sizes increase, the computational cost of pre-training escalates significantly. Moreover, end-to-end pre-trained models often lack flexibility, making it difficult to incorporate readily available unimodal pre-trained models, such as large language models (LLMs) (J. Li et al., 2023). This challenge underscores the need for architectures like the Unified Transformer, which can efficiently integrate multimodal data while mitigating the computational and flexibility constraints associated with traditional VLP methods.

### Modular Visual-Language Pretraining

Modular vision-language pretraining involves using pre-trained unimodal models (e.g., vision encoders and language models) and keeping them frozen while fine-tuning additional layers for vision-language alignment. Each module is optimised for its specific task or domain before integration into the larger system. This approach leverages existing models to reduce computational costs and enhance performance.

Modular vision-language pretraining involves using pre-trained unimodal models, such as vision encoders and language models, and keeping them frozen while fine-tuning additional layers for vision-language alignment. Each module is optimised for its specific task or domain before integration into the larger system. This approach leverages existing models to reduce computational costs and enhance performance.

Early work in this area employed a frozen object detector to extract visual features, with notable examples including Universal Image-Text Representation Learning (UNITER) (Y.-C. Chen et al., 2020), OSCAR (X. Li et al., 2020), and VinVL (P. Zhang et al., 2021). Recently, to bridge the gap between vision encoders and vision-language models, many state-of-the-art models freeze the language model and/or vision model for vision-to-language generation tasks. Notable models employing this approach include:

- Frozen (Tsimpoukelli et al., 2021): This model connects the final output vector of the vision encoder and linearly maps the output to  $D \times n$  channels, subsequently reshaping the result into a sequence of *n* embeddings. Here, *D* represents the token embedding dimension of the language model.
- Flamingo (Alayrac et al., 2022): This model enhances the integration of visual features by inserting new cross-attention layers into the large language model (LLM). These layers are specifically pre-trained to inject visual information into the model effectively.
- **BLIP-2** (J. Li et al., 2023): Bootstrapping Language-Image Pre-training (BLIP)-2 introduces a Querying Transformer (Q-Former), which consists of both an image transformer and a text transformer. These components jointly optimise three objectives that enforce the queries (a set of learnable embeddings) to extract visual representations most relevant to the accompanying text.

• LLaVA (H. Liu, Li, Wu, & Lee, 2023): LLaVA (Large Language and Vision Assistant) integrates modalities by introducing a projection layer between the open-set visual encoder of CLIP (Radford et al., 2021) and the language decoder Vicuna (Chiang et al., 2023). LLaVA also marks the first attempt to extend instruction-tuning to the language-image multimodal space, paving the way for the development of a general-purpose visual assistant.



Fig. 2.10 Comparison of VLM on 12 benchmarks (H. Liu, Li, Li, & Lee, 2023)

H. Liu, Li, Li, and Lee (2023) conducted a comprehensive evaluation of LLaVA 1.5 against state-of-the-art (SoTA) models, as depicted in Figure 2.4. Their findings revealed that LLaVA outperformed all other models in 11 out of 12 benchmarks and achieved a second-place ranking in the remaining benchmarks. This impressive performance underscores LLaVA's robustness and versatility.

Additionally, several derived works from LLaVA, such as LLaVA-Med (C. Li, Wong, et al., 2023), LLaVA-Gemma (Hinck et al., 2024), OMG-LLaVA (T. Zhang et al., 2024), and Math-LLaVA (Shi et al., 2024), have further demonstrated its adaptability and effectiveness across various domains and specialised tasks. These derivatives show LLaVA's capability to handle a wide range of applications, from medical imaging to mathematical problem-solving, highlighting its flexibility and broad applicability.

Given LLaVA's outstanding performance, relatively straightforward architecture, and proven ability to adapt to diverse tasks, it has been chosen as the project's foundational model.

# 2.5 Visual-Language Model as a Quantity Estimator

### 2.5.1 Visual Language Model as a Quantity Estimator

This project has a unique aspect as it utilises the VLM to estimate the flow rate of the 3D printing process. This requires using VLM as a numerical regressor. To ensure accurate regression, Mean Squared Error (MSE) (Equation 2.14) is typically employed. However, the loss function for most decoder-based LLMs, such as LLAMA (Touvron et al., 2023), GPT-3 (Brown et al., 2020), and T5 (Raffel et al., 2023), is cross-entropy (Equation 2.15), which optimises the probabilities of correctly classifying tokens in a fixed vocabulary. Using an LLM as a regressor effectively applies a classification loss function to solve a regression task. This also explains why in the VQA Challenge 2021 (Yadav et al., 2019), answering numerical questions had the worst performance out of all types of questions, as illustrated by Table 2.2. For example, numerical questions usually have a form like: "Count the number of people" or "How many shoes are there" (Goyal et al., 2017).

$$\mathscr{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} M_i \cdot (x_i - \hat{x}_i)^2$$
(2.14)

$$\mathscr{L}(y,\hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$$
(2.15)

where y is the true label vector (one-hot encoded) and  $\hat{y}$  is the predicted probability vector.

To address the challenge, Trott et al. (2018) trained a sequential counting mechanism with a reinforcement learning loss on the counting question subsets of VQA v2 and Visual Genome. They achieved a small increase in accuracy, but their method is not widely applicable to the traditional VQA framework. Models by Santoro et al. (2017) and Johnson et al. (2016) successfully counted on the synthetic CLEVR VQA dataset without bounding boxes and supervision of where the objects to count are. Y. Zhang et al. (2018) took a step further and proposed a neural network that leverages attention maps to allow robust counting, which improved the counting accuracy by 6.6% on the VQA v2 data. These methods have successfully improved the accuracy of numerical questions. However, they are not directly applicable to our project. The methods they propose primarily involve **counting objects** present in the image, whereas our VLM aims to **capture numerical information** and features

Rank	Participant team	yes/no	number	other	overall
1	PaLI-X - Google Research	96.78	74.14	79.46	86.06
2	PaliGemma-3B (finetune, 448px)	96.39	76.29	78.41	85.64
3	Zhipu AI	95.71	71.97	77.99	84.68
4	PaLI - Google Research	96.13	69.07	77.58	84.34
5	BEiT-3 (Microsoft)	96.43	73.63	75.92	84.18
6	mPLUG-single	94.83	69.82	77.02	83.62
7	MLLM A (MLLM-A-base)	94.69	69.46	75.80	82.94
8	XXW	94.85	72.24	74.15	82.52
9	CoCa - Google Brain	94.55	70.25	74.46	82.33
10	TiMix	94.37	67.28	74.82	82.10

Table 2.2 TOP 10 model accuracy  $Acc(ans) = min \{\frac{\#humans that said ans}{3}, 1\}$  in the VQA Challenge 2021 in August 2024 (start from 2021, no end date) (Yadav et al., 2019)

within the image and estimate their parameters (i.e. flow rate). This task is more complex and intricate than simple counting. As a result, this dissertation aims to close the gap in using VLMs to regress property values (not counting) from an input image.

### 2.5.2 Vision Encoder as a Quantity Regressor

#### Vision Encoder to Estimate Tyre-to-Road Friction

Prior work has employed vision encoders to regress properties from images. Vosahlik et al. (2021) utilised Convolutional Neural Networks (CNNs) combined with an Unscented Kalman Filter (Wielitzka et al., 2018) to estimate tyre-to-road interface friction parameters. Further advancing this concept, Ojala and Seppänen (2024) eliminated the need for sensor fusion via a Kalman filter by using ResNet50 (He et al., 2016) as the vision backbone. The authors simply added a regression head to the end of the vision backbone to output the friction estimation and its associated uncertainty.

#### Vision Encoder to Estimate Age

In the task of estimating a person's age from an image, several studies have focused on using Convolutional Neural Networks (CNNs) as feature extractors (Hou et al., 2017; Ojala & Seppänen, 2024). Recently, more attention has shifted towards Transformer-based approaches (Touvron, Cord, Sablayrolles, et al., 2021; Yuan, Hou, et al., 2021). Kuprashevich and Tolstykh (2023) proposed MiVOLO, which extracts facial and bodily features separately using Vision Transformers. These features are then processed by a regression head optimised

with a Weighted Mean Squared Error (WMSE) function (Y. Yang et al., 2021), effectively addressing the issue of imbalanced labels and features.

Overall, in both applications, it can be summarised that researchers typically add a regression head at the end of vision encoders, achieving high performance in estimating parameters inherent in images.

### 2.5.3 Large Language Model as Regressor

Recent research by Vacareanu et al. (2024) demonstrates that LLMs like GPT-4 and Claude 3 can perform linear and non-linear regression tasks using in-context learning. This work demonstrated that LLMs can generalise from a limited set of input-output pairs presented during inference, without requiring additional training or fine-tuning.

In their study, Vacareanu et al. (2024) utilised synthetic regression datasets for their experiments. For instance, in one of their linear regression tasks, they generated datasets where the relationship between the input and output was defined by a simple linear equation, such as y = wx + b, where w and b are coefficients. The training dataset consisted of several input-output pairs (e.g.,  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ ).

To test LLMs like GPT-4, they provided a few examples of input-output pairs directly as context during inference. The LLM then used these examples to predict the output for a new input without any additional training.

This is relevant in the context of our project, where the VLM's task of estimating flow rates in a 3D printing process can be seen as a regression problem. The empirical evidence provided by the work suggests that LLMs, when given appropriate in-context examples, can potentially achieve good performance in regression tasks.

There is a noticeable gap in leveraging VLMs to regress property values (e.g., flow rate) from images. A VLM could exploit the vision encoder's ability to capture detailed information while using the LLM to enhance interactivity, provide interpretable reasoning, and offer suggestions. Additionally, beyond estimating flow rate, a VLM could correct errors, assess extrusion quality, and provide explanations, which is beyond the capability of a pure vision encoder. This research aims to address this gap.

# 2.6 Chapter 2 Summary

In this chapter, we explored the integration of artificial intelligence in additive manufacturing for quality assurance.

We identified **Masked Autoencoder** (**MAE**) as a robust and efficient vision encoder due to its accuracy, simplicity, and stability. **LLaVA** was selected as the foundational model for its outstanding performance and adaptability across diverse tasks. To estimate numerical parameters (i.e. flow rate) from both image and text prompts, we decided to add a regression head to vision encoders.

These methodologies set the stage for the detailed implementation and experimental procedures in the following chapters.

# Chapter 3

# Methodology

This chapter details the design and implementation of the LLaVA-IntelliPrint system. It covers the architecture of the system, including the use of Masked Autoencoders (MAE) as the vision encoder and the integration of a regression head into the LLM. The chapter also describes the dataset used, the training process, and the evaluation metrics employed to assess system performance.

# 3.1 Large Language and Vision Assistant (LLaVA)

As explained in Chapter 2, LLaVA is selected as the foundational model due to its outstanding performance and adaptability across diverse tasks. This section introduces the architecture and properties of LLaVA, along with modifications made to our dataset to align with LLaVA's requirements.

# 3.1.1 Architecture of LLaVA

LLaVA (Large Language and Vision Assistant), developed by H. Liu, Li, Wu, and Lee (2023), is designed to understand and generate responses to both visual and language inputs. Its architecture consists of two primary components: the vision encoder and the language model.

The visual feature  $Z_v = g(X_v)$  is extracted from the input image  $X_v$ . A projection matrix W converts  $Z_v$  into language embedding tokens  $H_v$ , aligning with the word embedding space of the language model:

$$H_{\nu} = W \cdot Z_{\nu}, \quad \text{with} \quad Z_{\nu} = g(X_{\nu}) \tag{3.1}$$

Image patches are treated as language tokens and are directly fed into the language model to generate responses, as illustrated in Figure 3.1.

The vision encoder in LLaVA utilises the CLIP (Contrastive Language–Image Pre-training) visual encoder ViT-L/14 (Radford et al., 2021), which generates visual features from input images. The input image resolution is  $336 \times 336$  with a patch size of  $14 \times 14$ . The embedding dimension is 1024, resulting in  $\frac{336 \times 336}{14 \times 14} = 576$  patches, and thus image features represented by a vector of  $Z_{\nu} \in \mathbb{R}^{B \times 576 \times 1024}$ 

For the language model, LLaVA incorporates Vicuna (Chiang et al., 2023), an open-source language model renowned for its superior instruction-following capabilities, fine-tuned on top of LLaMA 2 (Touvron et al., 2023). The vision encoder and the language model are connected via a linear MLP layer. The input embedding dimension of each token for LLaMA 2 is 5120. Therefore, to map the vision representation into the same space as the language model, a two-layer projection layer is used, consisting of transformations from 1024 to 5120, as shown in Equation 3.2.

$$H_{\nu} = \operatorname{ReLU}(W_2 \cdot \operatorname{ReLU}(W_1 \cdot Z_{\nu} + b_1) + b_2)$$
(3.2)

where  $W_1 \in \mathbb{R}^{1024 \times 5120}, W_2 \in \mathbb{R}^{5120 \times 5120}, b_1$  and  $b_2$  are the biases

# 3.1.2 Training and Performance

LLaVA's training process involves a two-stage instruction-tuning procedure. Figures 3.1 and 3.2 illustrate the pre-training and fine-tuning stages, respectively, with frozen components indicated by snow symbols and active training components by fire symbols:

- 1. **Pre-training for Feature Alignment:** Using 595K image-text pairs from the filtered Conceptual Caption 3M dataset (Sharma et al., 2018), LLaVA is pre-trained to align visual features with the language model's word embeddings. This stage trains the projection matrix *W* while keeping the visual encoder and language model weights frozen (Figure 3.1).
- 2. **Fine-tuning End-to-End:** LLaVA is then fine-tuned on a multimodal instruction-following dataset (LLaVA-Instruct-158K) for tasks such as multimodal chatbot. This stage updates both the projection layer and the language model (Figure 3.2).

Both pretraining and fine-tuning use the Adam optimiser (Kingma & Ba, 2017), and the loss function is cross-entropy loss:

$$\mathscr{L}(y,\hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$$
(3.3)

where y is the true label vector and  $\hat{y}$  is the probability of each token logits.

LLaVA demonstrates impressive performance across various tasks. For instance, on the ScienceQA dataset, LLaVA achieved a state-of-the-art accuracy of 92.53%, surpassing previous models, including BLIP2 (J. Li et al., 2023), InstructBLIP (Dai et al., 2023) and Qwen-VL-Chat (Bai et al., 2023).



Fig. 3.2 LLaVA Finetune Stage (H. Liu, Li, Wu, & Lee, 2023)

# 3.2 LLaVA-IntelliPrint

As mentioned in previous sections, LLaVA uses Hugging face CLIP ViT-L/14 as its vision encoder and Vicuna as the language model. However, after training the off-the-shelf LLaVA on the 3D printing dataset, the performance was unsatisfactory (discussed further in the Experimental Results Section), with the estimated flow rate showing a mean absolute error (MAE) of 62.4%

CLIP ViT-L/14 was pre-trained on publicly available image-caption data. This was achieved through a combination of web crawling and using commonly used pre-existing image datasets such as YFCC100M (Thomee et al., 2016). These data mainly comprise media objects and

explain the rationale behind their creation, which do not typically include manufacturing images, and even less so 3D printing nozzles. Since the 3D printing dataset is so different from the pretraining images, the model may not have the appropriate capability to capture flow rate and extrusion quality information.

Therefore, pretraining and fine-tuning the vision encoder becomes the next immediate task. As mentioned in Chapter 2, due to the difficulty in collecting a labelled 3D printing dataset, self-supervised training is preferred to reduce the need for labelled data. Masked Autoencoder is selected as the vision encoder's evaluation metric.

To address this issue, we propose a four-stage training approach on the 3D printing dataset and a novel architecture to handle LLM for regression tasks:

# 3.2.1 Four-Stage Training Approach

As illustrated in Figure 3.3, the four-stage training process includes Pretraining the MAE ViT, Fine-tuning the MAE ViT, Pretraining LLaVA, and Fine-tuning LLaVA.



Fig. 3.3 LLaVA-IntelliPrint four-stage training

#### **Stage 1: Pretraining MAE ViT**

In this stage, the vision encoder is trained on an extensive 3D printing dataset using selfsupervised learning (SSL). This method enables the model to learn meaningful representations from the data without the need for labelled examples (Section 3.5.1). The primary objective is to minimize the Mean Squared Error (MSE) between the reconstructed and original image pixels, which is calculated as:

$$\mathscr{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} M_i \cdot (x_i - \hat{x}_i)^2$$
(3.4)

where  $x_i$  is the original pixel value,  $\hat{x}_i$  is the reconstructed pixel value, and  $M_i$  is the mask indicator function, which is 1 if the patch is masked and 0 otherwise.

#### **Stage 2: Fine-tuning MAE ViT**

Following pretraining, the vision encoder is fine-tuned on a labelled 3D printing dataset to specialize in flow rate estimation. (Section 3.5.2). The objective is minimising the Mean Squared Error (MSE) of flow rate predictions, calculated as:

$$\mathscr{L}(y,\hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
(3.5)

where  $y_i$  represents the true flow rate and  $\hat{y}_i$  represents the predicted flow rate.

#### Stage 3: Pretraining LLaVA

With the fine-tuned vision encoder, the next step is to pre-train the LLaVA model using our multimodal dataset (Section 3.4), which includes both visual and textual data related to 3D printing. The vision transformer (ViT) remains frozen during this stage. Visual features from the MAE ViT are projected into the word embedding space of the language model through a trainable projection multilayer perceptron (MLP). The goal is to align these visual features with the language model using the Cross-Entropy loss, defined as:

$$\mathscr{L}(t,\hat{t}) = -\sum_{i=1}^{n} t_i \log(\hat{t}_i)$$
(3.6)

where  $t_i$  denotes the true token and  $\hat{t}_i$  denotes the predicted probability of the token.

#### Stage 4: Fine-tuning LLaVA

The final stage involves fine-tuning the LLaVA model on a task-specific dataset (Section 3.4) to improve its performance on multimodal tasks relevant to 3D printing. During this phase, both the projection layer and the language model are updated. The fine-tuning process aims to optimise the model's ability to understand and generate accurate responses to multimodal

instructions by minimising the Cross-Entropy loss of the language model tokens, which is the same as Equation 3.6.

By following this four-stage training approach, we aim to significantly improve the model's performance in estimating flow rates in 3D printing processes and enhance its overall multimodal capabilities.

#### 3.2.2 LLM Regression Head

Building upon the Four-Stage Training Approach, an effective pipeline for achieving the dissertation's objectives, we propose a novel architectural enhancement for Vision-Language Models (VLMs). This enhancement introduces a dual-objective function that simultaneously optimises regression loss and language loss, thereby improving the accuracy of flow rate estimation. This architecture represents a significant advancement in using VLMs for regression tasks and can be extended to other fields with similar requirements.

As depicted in Figure 3.4, the architecture begins by embedding the vision and language inputs, which are subsequently processed by the LLAMA model to generate corresponding image and text outputs. Inspired by ViT, which employs a [CLS] token to aggregate image information, we integrate a similar concept into our architecture. Given the causal attention mechanism in language models, where only subsequent tokens can attend to preceding ones, we introduce a [SEP] token at the end of the image patch embeddings and before the text embeddings. This [SEP] token is designed to encapsulate all preceding image-related information.

The [SEP] token is then passed into a regression head responsible for predicting flow rates, which is optimised using the mean squared error (MSE) loss function. Concurrently, the language tokens are used in a cross-entropy loss function to refine the model's language predictions. These dual objectives are optimised simultaneously, resulting in a joint loss function, as shown in Equation 3.7:

$$\mathscr{L} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + \frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log \hat{p}_{i,c}$$
(3.7)

Here, *N* represents the number of samples,  $y_i$  is the true value for the *i*<sup>th</sup> sample,  $\hat{y}_i$  is the predicted value for the *i*<sup>th</sup> sample,  $y_{i,c}$  is a binary indicator (0 or 1) indicating whether class label *c* is the correct classification for sample *i*, and  $\hat{p}_{i,c}$  is the predicted probability of sample *i* belonging to class *c*.



Fig. 3.4 LLaVA-IntelliPrint Architecture

During inference, the output from the regression head is not used, as it does not influence the final text-based prediction. The hypothesis is that during training, the [SEP] token can attend to all preceding image patch tokens. Backpropagation through the regression head may enhance the model's numerical prediction by integrating this information. Additionally, since later language tokens attend to the [SEP] token, accurately representing numerical values could improve the language model's precision and knowledge, leading to better performance.

# **3.3 3D Printing Dataset**

To effectively implement the Four-Stage Training Approach and the novel LLM Regression Head architecture, the dataset must align with the foundation model's (i.e., LLaVA) requirements. Section 3.3.1 introduces our 3D Printing dataset, while Section 3.4 details the preprocessing steps taken to adapt it for the Four-Stage Training.

# 3.3.1 Dataset Introduction

The dataset used in this research was developed by the Computer-Aided Manufacturing Group at the Department of Engineering, University of Cambridge. It comprises over 500,000 labelled images capturing the material deposition process in extrusion additive manufacturing (AM). The images were generated using a Creality CR-20 Pro 3D printer equipped with an endoscope camera and a Raspberry Pi Model 4 B+ to record high-resolution images of the extrusion process. This dataset is crucial for training deep learning models for real-time control of 3D printing processes, enabling accurate prediction and correction of flow rates, thereby improving print quality and efficiency (Brion & Pattinson, 2022a).

Each image has a size of  $[350 \times 350]$ , and is labelled with metadata, specifically, the current material flow rate, obtained through a custom data acquisition system that integrates firmware metadata with real-time video. The dataset includes images sampled at 19 different flow rate levels, ranging from one-third to three times the optimal flow rate, ensuring a comprehensive representation of various printing conditions. Additionally, a natural language description of the 3D printing process is included. As shown in Figure 3.5, there are three main components in the text description:

- 1. Additive Manufacturing Description: Highlighted in blue, this section states the type of AM process (i.e., FDM).
- 2. **Printing Quality:** Highlighted in green, this section comprises two main pieces of information, which are the primary parameters to estimate:
  - Flow Rate: This percentage number defines the flow rate of the current printing process.
  - Extrusion Quality: There are three categories of extrusion quality: overextrusion, good extrusion, and under-extrusion, corresponding to a flow rate greater than, approximately equal to, and less than 100%, respectively.

3. **Explanation:** The last section, highlighted in grey, explores the potential reasons why this problem may have occurred and provides suggestions to alleviate it.



#### Image: CCR20PRO\_cylinder\_v1\_new-146293\_camera1.jpg

**Description:** This extrusion nozzle is pivotal in the Fused Deposition Modelling (FDM) additive manufacturing process, as depicted in the image. Analysis of the extrusion nozzle suggests a flowrate of 54 %, indicating a propensity towards under-extrusion. Such a phenomenon could arise from various sources, including filament contamination, worn-out nozzle components, or suboptimal printing parameters, all of which impede material flow and compromise print quality.



**Image:** split\_1.0mm\_CCR20PRO\_square\_cylinder\_4040\_x\_40\_v1\_new-188217\_camera1.jpg **Description:** Our focus is on the nozzle of an FDM 3D printer. Upon inspecting the extrusion nozzle, it becomes evident that the flowrate is 252 %. This higher-than-desired flowrate indicates over extrusion, potentially causing issues such as poor surface finish and dimensional inaccuracies.



**Image:** CCR20PRO\_circle\_in\_circle\_v2\_new-269659\_camera1.jpg **Description:** The extrusion nozzle seen here suggests the utilization of material extrusion, a technique frequently employed in additive manufacturing processes. The flowrate through this extrusion nozzle appears optimal, suggesting favourable extrusion conditions with a flowrate of 100 %. This steady flow promotes uniform material distribution, vital for intricate designs and complex geometries.

Fig. 3.5 Examples of the Dataset

# 3.3.2 Dataset Statistics

Out of the 500,000 data points, the dataset statistics are summarised in the following sections:

#### **Image Colour Statistics**

The RGB (Red, Green, Blue) values provide a quantitative measure of the average colour and variation in the images captured during the 3D printing process. As the colour of training images varies depending on the material colour, understanding these statistics helps in normalising the image for training.

- Dataset Image RGB Mean: (0.56101511, 0.57580587, 0.54373282)
- Dataset Image RGB Standard Deviation: (0.24505545, 0.2083447, 0.22679123)

The RGB values are scaled between 0-1. The mean values indicate a balanced colour distribution with a slight emphasis on green and blue channels. The standard deviations suggest moderate variability in colour, with the red channel showing the highest variation, which could be due to the material's reflectivity under the printing conditions. Figure 3.6 illustrates the proportion of dataset images based on their dominant colour, providing a comprehensive view of the colour variability present in the images.



Fig. 3.6 Dataset dominant colours distribution

#### **Extrusion Quality Distribution**

Table 3.1 presents the extrusion quality distribution, revealing a data imbalance, with 48.48% of prints exhibiting under-extrusion and 41.49% showing over-extrusion. Only 10.04% of the prints achieve good extrusion quality. This imbalance underscores the challenges in maintaining optimal flow rates, potentially leading to relatively lower classification accuracy for the Good Extrusion class.

Extrusion Quality	Count	Percentage
Over-Extrusion	207,432	41.49%
Good Extrusion	50,191	10.04%
Under-Extrusion	242,377	48.48%

 Table 3.1 Extrusion Quality Distribution

#### **Flow Rate Data Statistics**

The flow rate data provides insights into the material deposition process, which is critical for maintaining the quality of the 3D print. Table 3.2 displays a wide range of flow rates, with a mean of 121.38 and a large standard deviation of 76.05, indicating significant variability in the printing conditions.



Fig. 3.7 Flow Rate Data Distributions

Statistic	Value		
Median	100.00		
Mean	121.38		
Standard Deviation	76.05		
Minimum	33.00		
25th Percentile	56.00		
50th Percentile	100.00		
75th Percentile	168.00		
Maximum	300.00		

Table 3.2 Flow Rate Data Statistics

# 3.4 Training Dataset

Modifications and preprocessing have been applied to the 3D printing dataset to meet the requirements of LLaVA and Masked Autoencoder (MAE). This section details the changes made to the conversation data and the image augmentation techniques applied to the images, ensuring the model effectively learns and adapts to the specific characteristics of the data.

# 3.4.1 Conversation Format Modification

#### **Pretraining Dataset**

The pretraining dataset for LLaVA comprises one turn of conversation. The original LLaVA pretraining dataset is a subset of the CC3M dataset. To adapt the dataset for our purposes, we have developed a Python script that converts the 3D printing dataset as described in section 3.1 to the required format. The pretrain text description follows the format below:

```
Human: <image>\n Estimate the flow rate and + [Prompt]
GPT: [Answer]
```

To ensure that the model consistently predicts the critical parameter of flow rate, the text "Estimate the flow rate and" is added to each prompt. To adapt to various prompts, the [Prompt] in the dataset is a random sample from the list of instructions for image description. These instructions maintain the same meaning with natural language variance.

#### List of Instructions for Image Description

- Describe the following image in detail
- Provide a detailed description of the given image
- Give an elaborate explanation of the image you see
- Share a comprehensive rundown of the presented image
- Offer a thorough analysis of the image
- Explain the various aspects of the image before you
- Clarify the contents of the displayed image with great detail
- Characterize the image using a well-detailed description
- Break down the elements of the image in a detailed manner
- Walk through the important details of the image
- Portray the image with a rich, descriptive narrative
- Narrate the contents of the image with precision
- Analyze the image in a comprehensive and detailed manner
- Illustrate the image through a descriptive explanation
- Examine the image closely and share its details
- Write an exhaustive depiction of the given image
- Describe the manufacturing process.

An example of the pretrain data is given below:

### Example of Pretrain Data

Human: <image>\n Estimate the flow rate and offer a thorough analysis of the image GPT: The FDM 3D printer's nozzle can be seen.This extrusion nozzle exhibits signs of under-extrusion, suggesting that the flowrate may be 62 %. Under-extrusion commonly occurs when the feedrate is too low or the nozzle temperature is insufficient, resulting in inadequate material deposition.

#### **Fine-tuning Dataset**

Multi-turn instruction-following data is used to fine-tune the model. Based on the available information in the 3D printing dataset, the fine-tuned text description follows the format below:

```
Human: <image>\n Estimate the flow rate and + [Prompt]
GPT: [Answer]
Human: Estimate the flow rate
GPT: [Flow rate label]
Human: Describe the extrusion quality
GPT: [Extrusion quality label]
```

Similar to pertaining the [Promot] is randomly sampled from the list of instructions. To extract flow rate and extrusion quality labels from the free text, the following algorithm has been used and is capable of accurately extracting the information:

- Flow rate: The algorithm first splits the text into individual words. It identifies the flow rate by searching for numeric values that are before and after five words of the term "flow".
- Extrusion quality: The algorithm first splits the text into individual words. It looks for the word "extrusion" and checks nearby words for keywords indicating over-extrusion, under-extrusion, or good extrusion. Notice that several words have been used in the dataset to represent good extrusion. Therefore, a dictionary that has similar meanings to "good extrusion" based on the dataset has been built. If any of these words are found, it is classified as good extrusion.

```
Terms indicating good extrusion
```

```
"good", "proper", "optimal", "favorable", "consistent",
"effective", "reliable", "steady", "controlled"
```

An example of the finetune data is given below:

#### Example of Finetune Data

```
Human: <image>\n Estimate the flowrate and Illustrate the image
through a descriptive explanation
GPT: In our sights is the nozzle of a 3D printer, specifically an
FFF type.This extrusion nozzle exhibits signs of under-extrusion,
suggesting that the flowrate may be 66 %. Under-extrusion
commonly occurs when the feedrate is too low or the nozzle
temperature is insufficient, resulting in inadequate material
deposition.
Human: Estimate the flow rate
GPT: 66 %
Human: Describe the extrusion quality
GPT: under-extrusion
```

### 3.4.2 Image Pre-Processing

In this study, image preprocessing is a crucial step to ensure the efficacy of the Masked Autoencoder model. Proper preprocessing enhances the model's ability to learn and generalise from the input data. The following steps outline the specific transformations applied to the images before they are fed into the model:

- **Resize Images:** All images from the 3D printing dataset have a dimension of [350 × 350] and are downscaled to a standard input size of [224 × 224]. This standardisation ensures consistent input dimensions for the model.
- **Interpolation:** Use bicubic interpolation for resizing images. Bicubic interpolation can be represented as:

$$I'(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} w(i,j) \cdot I(x+i-1,y+j-1),$$
(3.8)

where *I* is the input image, *I'* is the resized image, *i* and *j* are incices that iterate over the  $4 \times 4$  neighbourhood and w(i, j) are the bicubic interpolation weights computed as:

$$w(i,j) = (1 - |i|/a) (1 - |j|/a)$$
(3.9)

with *a* being a parameter controlling the smoothness of the interpolation.

- **Random Flipping:** Apply random horizontal flipping to the images. This augmentation technique helps in making the model invariant to the orientation of objects in the images.
- Normalisation: Normalise the images using pre-defined mean μ and standard deviation σ values for each channel. As presented in Section 3.1.2, the dataset has a mean of (0.561, 0.576, 0.544) and a standard deviation of (0.245, 0.208, 0.227). The normalisation process is represented as:

$$I' = \frac{I - \mu}{\sigma},\tag{3.10}$$

ensuring that the pixel values are consistent, which improves the model's performance. The effect of the image processing applied to a raw image can be observed below:



Fig. 3.8 Original Image



Fig. 3.9 Image after preprocessing

By applying these transformations, the dataset becomes more robust, mitigating overfitting and ensuring that the Masked Autoencoder can generalise well to unseen data. The impact of each transformation on the dataset's quality and the model's performance is critically analysed in subsequent sections.

Some other image processing techniques, such as Random Erasing and PCA Colour Augmentation, have also been attempted.

- **Random Erasing:** Random erasing introduces the removal of parts of an image with a certain probability (Z. Zhong et al., 2017).
- **PCA Colour Augmentation:** Following Brion and Pattinson (2022a), PCA colour augmentation was attempted. This technique adjusts the principal components of the image's colour distribution to introduce slight variations.

However, during the ablation studies of image preprocessing techniques, they demonstrated a negative effect on flow rate estimation and were subsequently removed in the MAE training, which will be presented in Chapter 4.2.

# 3.5 Masked Autoencoder (MAE)

The first two stages of the proposed four-stage training involve pretraining and fine-tuning the MAE vision encoder. This section presents the technical details of these two stages.

Masked Autoencoders (MAE) (He et al., 2021) adapt the autoencoder framework, which includes an encoder to represent images in a latent space and a decoder to reconstruct the original signal from the latent representation. This concept extends from denoising autoencoders, which extract important features and reduce noise by reconstructing a clean version of the input (Vincent et al., 2008). MAE employs an asymmetric design, allowing the encoder to operate only on the partial, observed signal (without mask tokens), while a lightweight decoder reconstructs the full signal from the latent representation and mask tokens. This ensures that most of the learning is done by the encoder, with the decoder being less complex and discarded when using the Vision Transformer (ViT) as an image encoder.



Fig. 3.10 Masked Autoencoder Architecture (He et al., 2021)

### **3.5.1** Pretraining Phase

The first training phase is the pretraining phase, which uses the self-supervised learning (SSL) technique to learn image representations. The architecture of MAE is shown in Figure 3.10. There are three key components:

• Masking: Similar to the standard ViT, input images are divided into non-overlapping patches. For the commonly used MAE-ViT-Large Patch 16, images are resized to  $[224 \times 224]$  pixels, and each patch has a size of  $[16 \times 16]$ . Therefore, there are a total of 196 patches, as shown in Equation 3.11. A [CLS] token is also added to the start of all patches, aiming to capture the information of the whole image. Thus, there are 196 + 1 = 197 patches. A certain ratio of patches are randomly masked, with a masking ratio of 75% showing the best performance.

$$\frac{224 \times 224}{16 \times 16} = 196 \text{ patches}$$
(3.11)

• MAE Encoder: The encoder is a ViT, which embeds patches using a linear projection with added positional embeddings and then processes the resulting set through a series of Transformer blocks. It only encodes the unmasked patches to extract information. The authors have provided several standard architectures of ViT in their official code, as detailed in Table 3.3.

ViT	Image Size	Patch Size	Embedding Dim	Depth	Heads
ViT Tiny Patch 16	$224 \times 224$	$16 \times 16$	192	12	3
ViT Small Patch 16	$224 \times 224$	$16 \times 16$	384	12	6
ViT Base Patch 16	$224 \times 224$	$16 \times 16$	768	12	12
ViT Large Patch 16	$224 \times 224$	$16 \times 16$	1024	24	16
ViT Huge Patch 14	$224 \times 224$	$14 \times 14$	1280	31	16

Table 3.3 Different types of ViT implementations in MAE

• **MAE Decoder**: The input to the MAE decoder is the full set of tokens, including both encoded visible patches and mask tokens.

The masked sequence  $\mathbf{Z}_m$  is then fed into a decoder to reconstruct the original image. The decoder first upsamples the sequence by filling in the masked positions with a learnable embedding  $\mathbf{e}_{\text{mask}} \in \mathbb{R}^D$ , *D* is the embedding dimension, creating a sequence  $\mathbf{Z}_d$  that matches the original length *N*:

$$\mathbf{Z}_d = \mathbf{Z}_m + (1 - \mathbf{M}) \odot \mathbf{e}_{\text{mask}}$$
(3.12)

Where  $\mathbf{M}$  is the mask indicator matrix. It is a binary matrix where each element indicates whether a corresponding token in the original sequence was masked.

This sequence is then processed by the decoder, which consists of multiple layers of transformer blocks:

$$\mathbf{\hat{x}}_i = \text{Decoder}(\mathbf{Z}_d) \tag{3.13}$$

where  $\mathbf{\hat{x}}_i$  represents the reconstructed patch embeddings.

The decoder predicts the pixel values for each masked patch. The loss function for reconstruction is the mean squared error (MSE) between the reconstructed and original images in the pixel space. (Equation 3.4)

#### **Pretraining Implementations**

He et al. (2021) provided the pretrained checkpoint that has been trained on ImageNet-1K (Russakovsky et al., 2015). To leverage the capabilities of transfer learning, our pretraining process initialized the model with the parameters provided by the authors and subsequently trained the model on our dataset. This approach allows the model to benefit from the extensive training on a large dataset like ImageNet-1K, improving generalization, convergence speed and performance on our specific task.

#### **3.5.2** Fine-tuning Phase

The second stage is the fine-tuning phase. In this phase, the decoder is discarded, leaving only the ViT to output a vector representing the input image. There are two ways to obtain the learned vector:

• Use the [CLS] token

- A [CLS] (Class Token) is added to the start of image patches in Vision Transformers. This token aggregates information for classification tasks (Dosovitskiy et al., 2021).
- Pooling: Average/Max over all patches

 Instead of feeding the [CLS] token into the MLP head, average and max pooling (Equation 3.14) have also been widely utilised in recent transformers and have demonstrated good performance in classification tasks (Z. Liu et al., 2021; Marin et al., 2023).

$$Z_{\rm avg} = \frac{1}{N} \sum_{i=1}^{N} Z^{(i)}$$
(3.14)

where  $Z^{(i)} \in \mathbb{R}^D$  represents the embedding of the *i*-th patch.

Both the [CLS] token and pooling techniques are experimented with in this dissertation. An unmasked image is fed into the encoder. The learned vector is then input into an MLP regression head that outputs the flow rate prediction. In our case, since the patch embedding size for ViT-L that we use is 1024, the regression head has a structure of

$$\mathbf{y} = \operatorname{ReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \tag{3.15}$$

Where  $\mathbf{W}_1 \in \mathbb{R}^{1 \times 1024}$  is the weight matrix,  $\mathbf{b}_1 \in \mathbb{R}^1$  is the bias term.

The objective is to reduce the mean squared error (MSE) (Equation 3.16) and is optimised using the Adam optimiser.

$$\mathscr{L}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
(3.16)

where  $y_i$  represents the true flow rate and  $\hat{y}_i$  represents the predicted flow rate.

#### **Fine-tuning Implementations**

During the code implementation, the scale of the flow rate prediction *y* ranged from 0 to 2 by inputting 1000 images into the Vision Transformer (ViT) without any optimisation. Since the actual flow rate labels *y* range from 33 to 300, a min-max scaling has been applied to normalise the label data. This enhances convergence speed and numerical stability by ensuring that the data is within a consistent range, which benefits the optimisation algorithm. The min-max scaling is calculated using Equation 3.17.

$$x' = \frac{(x - \text{original}\_\min) \cdot (\text{target}\_\max - \text{target}\_\min)}{\text{original}\_\max - \text{original}\_\min} + \text{target}\_\min$$
(3.17)



Fig. 3.11 MAE Fine-tune Architecture

Where original\_min, original\_max, target\_min, target\_max are the min/max values of the original data range and the min/max values of the target data range respectively.

# **3.6 Evaluation Metrics**

This section introduces the evaluation methods for both the Masked Autoencoder (MAE) Vision Transformer (ViT) and the Visual Language Model (VLM). These evaluation metrics are applied consistently across all experiments to enable a fair comparison of different models.

# 3.6.1 Masked Autoencoder Performance Evaluation

### **Pretrain - Image Reconstruction**

The primary quantitative metric used to measure the reconstruction quality of images is the Mean Squared Error (MSE). The MSE evaluates the average squared difference between the original and reconstructed images, providing a clear measure of reconstruction accuracy. The formula for MSE, considering the masked patches, is given by Equation 3.16. Additionally, qualitative evaluation involves visually inspecting the reconstructed images.

#### **Finetune - Flow Rate Estimation**

For flow rate estimation, we employ the Mean Absolute Error as our evaluation metric. MAE provides a clear measure of the average magnitude of errors in the predictions, which is particularly useful for understanding how close the predicted flow rates are to the true labels. The formula for MAE is given by:

Mean Absolute Error 
$$=$$
  $\frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|$  (3.18)

where  $\hat{y}_i$  and  $y_i$  represent the predicted and true flow rates, respectively, and *n* is the number of observations.

### 3.6.2 Visual Language Model Performance Evaluation

We evaluate the performance of the VLM on the 3D printing task based on four dimensions: Flow Rate Estimation, Extrusion Quality Estimation, Language Consistency, and Text Generation Quality.

#### **Flow Rate Estimation**

Flow rate is identified by searching for numeric values within five words of the term "flow". The mean absolute error of flow rate estimation is then reported using the same MAE formula described previously (Equation 3.18).

#### **Extrusion Quality Estimation**

To systematically measure the performance of the VLM in classifying extrusion quality (i.e., over-extrusion, under-extrusion, and good-extrusion), the evaluation is extracted from the generated text by locating the keyword "extrusion" and checking nearby words for keywords indicating its quality. The overall accuracy, precision per class, recall per class, and F1 score per class are calculated. These metrics provide a comprehensive evaluation of the model's classification performance by considering both the positive and negative predictions. The formulas for accuracy, precision, recall, and F1 score are given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
(3.19)

$$Precision = \frac{TP}{TP + FP}$$
(3.20)

$$\operatorname{Recall} = \frac{TP}{TP + FN} \tag{3.21}$$

F1 Score = 
$$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
 (3.22)

where *TP*, *FP*, and *FN* represent the true positives, false positives, and false negatives, respectively.

#### Language Consistency

Language Consistency evaluates whether the predicted flow rate aligns with the described extrusion quality in the generated text. Specifically, it checks if a flow rate below, equal to, or above 100% is correctly described as under, good, or over-extrusion, respectively. For example, the second text below is classified as inconsistent because it describes the extrusion quality as "good" despite a flow rate greater than 100%.

- **Consistent text**: The extrusion nozzle seen here suggests the utilisation of material extrusion, a technique frequently employed in additive manufacturing processes. Upon inspecting the extrusion nozzle, it becomes apparent that the flowrate is **300** %. This higher-than-desired flowrate suggests **over extrusion**, potentially causing issues such as poor surface finish and dimensional inaccuracies
- Inconsistent text: Upon inspection of the extrusion nozzle, it seems that the flowrate is **112** %. This consistent flowrate indicates **good extrusion**, ensuring proper material deposition and layer adhesion, ultimately contributing to the structural integrity of the printed object.

The percentage of consistent predictions is reported. The consistency can be mathematically evaluated as follows:

$$Consistency = \frac{\text{Number of consistent predictions}}{\text{Total number of predictions}} \times 100\%$$
(3.23)

#### **Text Generation Quality**

To assess the natural language generation quality, we use BERTScore. Traditional metrics like BLEU (Papineni et al., 2002) and METEOR (Banerjee & Lavie, 2005) provide quantitative measures for text generation by evaluating n-gram overlap and flexible matching, respectively. However, they often fail to capture the full semantic context, making them less suitable for assessing nuanced text generation. In contrast, BERTScore (T. Zhang et al., 2020)

evaluates semantic similarity using pre-trained BERT embeddings, offering a more robust, context-aware assessment. BERTScore calculates precision, recall, and F1 score as follows:

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^{\top} \hat{x}_j$$
(3.24)

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^\top \hat{x}_j$$
(3.25)

$$F_{\text{BERT}} = 2 \times \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$
(3.26)

where *x* represents the set of tokens in the reference text,  $\hat{x}$  is the set of tokens in the generated text, and  $x_i^{\top} \hat{x}_j$  denotes the cosine similarity between the BERT embeddings of tokens  $x_i$  and  $\hat{x}_j$ . A BERTScore typically ranges from 0.85 to 0.90 or higher, indicating a close semantic match between the generated and reference texts. This is crucial for ensuring the practical usability of the model's text outputs.

By applying these rigorous evaluation metrics, we ensure a comprehensive assessment of both the MAE ViT and VLM models. These metrics not only provide quantitative measures of performance but also offer qualitative insights that are essential for the practical deployment of these models in real-world applications.

# **3.7 Chapter 3 Summary**

Chapter 3 outlines the methodology behind our proposed system, LLaVA-IntelliPrint, the first vision-language model designed for real-time quality assurance in 3D printing processes. Built on the foundation of the LLaVA model, the system integrates a large language model (LLM) with a Vision Transformer (ViT) and adopts a four-stage training approach to enhance its ability to identify and correct defects during additive manufacturing. A novel regression head is introduced to enable the LLM to perform numerical regression tasks, such as estimating flow rates from visual inputs. The methodology also covers comprehensive dataset preparation, including preprocessing steps and the use of a 3D printing dataset for training. Additionally, the Masked Autoencoder (MAE) is employed as a vision encoder to ensure accurate flow rate prediction and error detection. Evaluation metrics, including mean absolute error for flow rate estimation and VerScore for text generation quality, are used to assess the system's performance.
## **Chapter 4**

## **Experiments & Results**

This section details the experimental setups and results for LLaVA-IntelliPrint, evaluated on the test set of the 3D printing dataset. We conducted two primary sets of experiments: one focused on the MAE Vision Encoder and the other on the LLaVA-IntelliPrint Vision-Language Model. Additionally, we present ablation studies that investigate the impact of each model component. Finally, we discuss the model's performance in the context of its relevance to the initial use case, highlighting its practical applicability.

## 4.1 Experiment Setup

## 4.1.1 Train Test Split

As mentioned in Section 3.3, the dataset used in this project consists of 500K labelled imagetext pairs. An 8-1-1 train-validation-test split was implemented, resulting in the following dataset sizes:

Model	Data Inputs	Train Size	Validation Size	Test Size			
MAE	Image + Flow Rate Labels	400K	50K	50K			
LLaVA-IntelliPrint	Image + Text References	400K	50K	1K			
Table 4.1 Detect Distribution for Demonstrate							

 Table 4.1 Dataset Distribution for Experiments

Notice that performing 50,000 test data point inferences on LLaVA-IntelliPrint would require approximately 40 GPU hours on a single NVIDIA A100. Due to computational constraints, 1,000 data points were sampled from the 50,000 test data.

## 4.1.2 MAE Vision Encoder Experiments Overview

#### **Training Parameters**

The experiments conducted with the MAE Vision Encoder included several ablation studies to optimise its performance. Both the pretraining and fine-tuning phases were carried out using 4 NVIDIA A100 GPUs. The specific parameters utilised during pretraining and fine-tuning are detailed in Table 4.2 and Table 4.3, respectively.

		Parameter	Value	
Parameter	Value	Batch Size	256	
Batch Size	64	Model	ViT Large Patch16	
Model	ViT Large Patch16	Epochs	20	
Masking Ratio	0.75	Base Learning Rate	1.5e-3	
Epochs	20	Layer Decay	0.75	
Base Learning Rate	1.5e-3	Weight Decay	0.05	
Weight Decay	0.05	Drop Path Rate	0.2	
Image Size	224	Image Size	224	
Table 4.2 Key Traini	ng Parameters for	Gradient Clipping	1.0	
MAE Pretrain		Table 4.3 Key Training Parameters for		
		MAE Finetune		

#### **Ablation Studies**

Two ablation studies were conducted to explore various aspects of the Masked Autoencoder (MAE). The first ablation study focuses on evaluating the impact of different configurations within the **MAE architecture**. The second study examines the effectiveness of various **image augmentation techniques** applied within the MAE framework.

### 4.1.3 LLaVA-IntelliPrint VLM Experiments Overview

#### **Training Parameters**

The experiments conducted with the LLaVA-IntelliPrint Vision Language Model included several ablation studies and model enhancement techniques to optimise its performance. Both the pretraining and fine-tuning phases were carried out using 4 NVIDIA A100 GPUs. The specific parameters utilised during pretraining and fine-tuning are detailed in Table 4.4 and Table 4.5, respectively.

#### 4.1 Experiment Setup

Parameter	Value					
Num Train Epochs	1					
Per Device Train Batch Size	32					
Per Device Eval Batch Size	4					
Gradient Accumulation Steps	1					
Learning Rate	1e-4					
Warmup Ratio	0.03					
LR Scheduler Type	cosine					
Model Max Length	2048					
Table 4.4 Key Pretraining Parameters for						
LLaVA-IntelliPrint						

Parameter	Value				
Num Train Epochs	1				
Per Device Train Batch Size	16				
Per Device Eval Batch Size	4				
Gradient Accumulation Steps	1				
Learning Rate	2e-5				
Warmup Ratio	0.03				
LR Scheduler Type	cosine				
Model Max Length	2048				
Table 4.5 Key Finetuning Parameters for					

## LLaVA-IntelliPrint

#### **Ablation Studies**

Table 4.6 summarises the 11 ablation studies conducted on various modules and architectures of LLaVA-IntelliPrint. **Experiment 0** serves as the baseline, demonstrating the performance of fine-tuning the 3D printing dataset on the out-of-the-box LLaVA model without any modifications. **Experiments 1-6** investigate the effectiveness of different training components within LLaVA-IntelliPrint. **Experiments 7-8** assess the impact of various numerical tokenisation approaches. Finally, **Experiments 9-10** evaluate the influence of the novel architecture by adding a regression head during training.

	Pretrain	Fine-tune	Train	Numeric	Regression
	MAE	MAE	LLaVA	Tokenisation	Head
Experiment 0			$\checkmark$		
Experiment 1		$\checkmark$	$\checkmark$		
Experiment 2	$\checkmark$		$\checkmark$		
Experiment 3	$\checkmark$	$\checkmark$	$\checkmark$		
Experiment 4	$\checkmark$	$\checkmark$			
Experiment 5		$\checkmark$			
Experiment 6	$\checkmark$				
Experiment 7	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	
Experiment 8			$\checkmark$	$\checkmark$	
Experiment 9	$\checkmark$	$\checkmark$	$\checkmark$		$\checkmark$
Experiment 10			$\checkmark$		$\checkmark$

Table 4.6 LLaVA-IntelliPrint Ablation Study

#### **Data Efficiency Optimisation**

Finally, we performed a data efficiency optimisation to determine whether the entire 500K dataset is necessary for the model to converge.

## 4.2 MAE Vision Encoder Performance

## 4.2.1 Best Performing Model

The best-performing model is the ViT-Large Patch 16, which is both pre-trained and finetuned on the 3D Printing dataset. This model utilised the [CLS] token connected with a regression head to achieve superior performance in regression tasks. The image preprocessing steps included resizing, interpolation, random flipping, and normalisation.

As some examples demonstrated in Figure 4.1, this model achieved a Mean Absolute Error (MAE) of 6.49%, significantly outperforming the benchmark MAE of 8.64% (Margadji et al., 2024). This improvement underscores the effectiveness of adding a regression head for MAE for flow rate estimation in 3D printing applications.



Reference: 78 Prediction: 78



Reference: 200 Prediction: 202



Reference: 100 Prediction: 100

Fig. 4.1 Examples of MAE Flow rate estimation.

## 4.2.2 Ablation Studies on MAE Architecture

This ablation study investigates how the Vision Transformer (ViT) size, pretraining strategies, and image feature representation methods affect the MAE model's ability to reconstruct and estimate flow rates from 3D printing images.

The ablation study aims to explore three key areas:

- 1. Pretraining: The impact of pertaining stage is evaluated by comparing two approaches:
  - Setup A: Pretraining on the 3D Printing Dataset on top of the ImageNet weights, followed by finetuning.

- **Setup B**: Use the publicly available pretraining weight on ImageNet and only fine-tune it on the 3D Printing Dataset.
- 2. Feature Representation: The performance differences between using the [CLS] token and the pooling technique as the image feature vector for flow rate regression are examined.
- 3. **Model Size**: The study compares the performance of different ViT model sizes—ViT-Small, ViT-Base, and ViT-Large—to assess how model capacity influences the results.

Table 4.7 presents the results for Training Setup A using different ViT sizes after pretraining and finetuning on the 3D printing dataset. The corresponding image reconstruction results are shown in Figure 4.2. Similarly, Table 4.8 shows the results for Training Setup B, with finetuning based on ImageNet pretrained weights. The reconstruction results are shown in Figure 4.3.

ViT Size	Stage	MAE (%)	Validation Loss	STD
	Pre-train on 3D Printing Data	-	0.0511	0.0026
Small-16	Fine-tune (Pooling)	39.47	-	7.50
	Fine-tune (CLS Token)	27.83	-	3.14
Base-16	Pre-train on 3D Printing Data	-	0.0460	0.0025
	Fine-tune (Pooling)	38.90	-	5.22
	Fine-tune (CLS Token)	25.49	-	2.03
Large-16	Pre-train on 3D Printing Data	-	0.0374	0.002
	Fine-tune (Pooling)	36.82	-	5.36
	Fine-tune (CLS Token)	24.24	-	5.26

Table 4.7 Training Setup A - MAE flow rate regression results: Pretraining on 3D Printing Data; Finetuning on the 3D Printing pre-trained weights.



Fig. 4.2 MAE ViT-L reconstruction of 3D Printing Data set based on weights pretrained on the 3D printing dataset.

ViT Size	Stage	MAE (%)	std
	Fine-tune (Pooling)	40.32	6.10
Small-16	Fine-tune (CLS Token)	28.23	4.14
	Fine-tune (Pooling)	39.15	6.72
Base-16	Fine-tune (CLS Token)	25.82	2.23
	Fine-tune (Pooling)	36.44	6.56
Large-16	Fine-tune (CLS Token)	24.93	5.59

Table 4.8 Training Setup B - MAE flow rate regression results: Finetuned on publicly available pretrained weights from ImageNet.



Fig. 4.3 MAE ViT-L reconstruction of 3D Printing Data set based on weights pretrained on ImageNet.

#### **Analysis of Pretraining**

Pretraining with ImageNet (Training Setup B) appears to have better reconstruction capabilities, possibly due to the generalised nature of ImageNet, which contains a diverse set of natural images. However, compared with the natural images, 3D printing nozzle images are out of distribution, which may introduce noise during finetuning for this specific task, as they differ significantly from the images in ImageNet. Despite this, it is important to note that a better reconstruction quality does not necessarily correlate with better feature representation, as suggested by He et al. (2021). This is evidenced by the lower Mean Absolute Error (MAE) observed in Training Setup A (24.24%) than in Training Setup B (24.93%), which indicates that pretraining on the 3D Printing Dataset better facilitated the specific task of flow rate estimation.

#### **Analysis of Image Feature Representation**

In terms of image feature representation, the [CLS] token consistently outperformed the pooling technique across all ViT sizes, yielding lower MAE values. The [CLS] token is

specifically designed to encapsulate the global representation of the entire image through self-attention in the inner network. In comparison, Pooling averages information across all image patches, potentially leading to information loss.

#### **Analysis of Model Size**

The analysis of model size demonstrates that the ViT-Large model achieved the lowest MAE, suggesting that its greater capacity allows it to capture more complex patterns and features from the data. The enhanced performance of the ViT-Large model is likely due to its superior ability to represent the intricate details within the 3D printing images, which are crucial for accurate flow rate estimation. In comparison, smaller models, even though useful, might not have the ability to capture all intricate information, which yields a slightly worse performance. It also indicates that our 3D Printing dataset is complex enough that it did not underfit ViT-Large.

#### Section 4.2.2 Summary

In summary, the ablation studies underscore the importance of both pretraining and finetuning on the 3D Printing Dataset, utilising the [CLS] token for feature representation and leveraging the ViT-Large model for optimal performance in flow rate estimation tasks.

### 4.2.3 Ablation Studies on Image Augmentation Methods

Despite identifying the best architecture in terms of Mean Absolute Error (MAE), the optimal performance, with an MAE of 24.24%, is still significantly higher than the benchmark of 8.64% set by Margadji et al. (2024), who employed Convolutional Neural Networks (CNNs) for the regression task. Given that Vision Transformers (ViTs) are generally considered more powerful models, this discrepancy prompted an investigation into various image augmentation methods to enhance model performance.

We evaluated random flipping, random erasing, and PCA colour augmentation as augmentation techniques. Other preprocessing methods, such as resizing, interpolation, and normalisation, were kept unchanged due to their importance in ensuring consistent input data dimensions and distribution. Each augmentation technique was individually tested to assess its impact on the model's performance, focusing on the ViT-Large model, which demonstrated the best performance in the previous analysis. The results are presented in Table 4.9.

Augmentation Technique	Training Setup A	Training Setup B
Random Flipping	$6.99 \pm 0.96\%$	$6.49 \pm 0.80\%$
Random Erasing	$24.14 \pm 0.61\%$	$24.23 \pm 0.47\%$
PCA Colour Augmentation	$28.05 \pm 0.72\%$	$25.13 \pm 0.54\%$

Table 4.9 Comparison of Regression MAE (Mean ± Std) for Different Augmentation Techniques

#### Analysis of the Results

The results suggest that selective augmentation techniques can significantly influence model performance:

- **Random Flipping**: The model achieved notably lower MAEs of 6.99% and 6.49% when random flipping was **the only augmentation** method used. This indicates that introducing variability through image orientation and not implementing the other two techniques are beneficial.
- **Random Erasing**: The results show that **not using random erasing** led to improved performance. While random erasing can reduce overfitting by discouraging the model from relying on specific image regions. For 3D printing images, where critical information is concentrated in specific areas (e.g., the nozzle), random erasing of these areas will cause the model to fail to extract useful flow rate representations. That is why removing such a technique reduces prediction errors.
- **PCA Colour Augmentation**: The model performed better when PCA colour augmentation was omitted. Although PCA colour augmentation can improve generalisation by altering colour distributions, in this scenario, it negatively impacted the model's ability to accurately estimate flow rates. Avoiding this technique helped preserve crucial image features necessary for precise predictions.

#### Section 4.2.3 Summary

In conclusion, the study achieved state-of-the-art performance in flow rate estimation (6.49% MAE) by employing a two-step approach: pretraining on the 3D Printing Dataset using ImageNet weights, followed by finetuning on ImageNet data with the ViT-Large architecture. The image preprocessing methods utilised resizing, interpolation, random flipping, and normalisation, which collectively contributed to optimising model performance while ensuring consistency and reliability.

## 4.3 LLaVA-IntelliPrint VLM Performance

After identifying the optimal Vision Encoder setup for flow rate estimation, we integrated it into our VLM—LLaVA-IntelliPrint. The performance is presented and discussed in this section.

## 4.3.1 Best Performing Model

The best-performing model is based on the setup from Experiment 3, which employs a Masked Autoencoder (MAE) pre-trained and fine-tuned ViT-Large Patch 16 model. This is followed by training on LLaVA with a regression head added to the language model. This model achieved impressive results in flow rate estimation mean absolute error (MAE) of 12.02%, a classification accuracy of 90%, language consistency of 94%, and text generation quality, as evidenced by a BERTScore F1 of 0.91.

This model is the first Vision-Language Model (VLM) applied to the 3D printing domain. It successfully demonstrated the ability to perform not only regression and classification tasks but also to generate accurate free-text explanations. This establishes a robust benchmark, proving that VLMs can be effectively utilised in 3D printing, and sets a foundation for advancing machine learning applications in this field. An example of the model's performance is shown in Figure 4.4.



Fig. 4.4 Example of LLaVA-IntelliPrint

- Prompt: Estimate the flow rate and Describe the manufacturing process.
- Label: Material extrusion, a common technique in additive manufacturing, is hinted at by the extrusion nozzle visible in this close-up.From this extrusion nozzle, there are clear signs of over-extrusion, likely operating at 283 % flow rate. Over-extrusion,

caused by a high feed rate or excessive temperature, may result in problems like nozzle clogging or unsatisfactory surface finish.

• **Prediction**: The focal point of this image is the extrusion nozzle, which suggests the operation of a material extrusion process, possibly within additive manufacturing. There are clear indications of over-extrusion from this extrusion nozzle, potentially functioning at 283 % flow rate. Over-extrusion can occur due to a high feed rate or excessive temperature, leading to problems like nozzle clogging or inadequate surface finish.

The LLaVA-IntelliPrint model demonstrates remarkable accuracy in identifying the extrusion nozzle as the focal point of the image, which is critical in additive manufacturing processes. The consistent identification of the extrusion nozzle as the focal point across all predictions highlights its central role in determining the quality and outcome of material extrusion. The model correctly diagnoses over-extrusion, quantified at 283% flowrate, and links this to potential causes such as high feedrate or excessive temperature. This precise interpretation, coupled with the use of domain-specific terminology, underscores the model's advanced understanding of the manufacturing process and its potential utility in quality control and process optimisation.

In the next sections, we will conduct ablation studies to analyse the effect of each training and architectural component.

## 4.3.2 LLaVA-IntelliPrint Ablation Study: Experiment 0

As illustrated in Table 4.10, Experiment 0 is served as a baseline which was carried out by fine-tuning the original LLaVA model, without any modifications, on our 3D printing dataset. This experiment serves as a reference point for assessing the effectiveness of the LLaVA-IntelliPrint training pipeline and architecture.

	Pretrain	Fine-tune	Train	Numeric	Regression	
	MAE	MAE	LLaVA	Tokenisation	Head	
Experiment 0			$\checkmark$			
Table 4.10 LLaVA-IntelliPrint Baseline Setup						

The baseline model yielded a mean absolute error (MAE) of 62.4 %. This relatively high error suggests that the out-of-the-box LLaVA model is not well-suited to the 3D printing dataset. The suboptimal performance could be attributed to the model not being pre-trained

62

Experiment Number	Mean Absolute Error	Consistency Percentage	Extrusion Quality Accuracy	Extrusion Quality Precision	Extrusion Quality Recall	Extrusion Quality F1	BERTScore Precision	BERTScore Recall	BERTScore F1
0	62.4	1.0	0.51	[0.97 0.2 0.45]	[0.21 0.13 0.96]	[0.34 0.17 0.63]	0.91	0.91	0.91

Table 4.11 LLaVA on 3D Printing Dataset Baseline

on data that closely resembles the 3D printing domain, making it challenging for the model to capture the intricate relationships between the input features and the output flow rate.

#### 4.3.3 LLaVA-IntelliPrint Ablation Study: Experiment 1-6

To systematically analyse and understand the contributions of different modules to the overall performance of the LLaVA-IntelliPrint VLM, 6 experiments were proposed. These experiments focus on different stages of the vision encoder and language model training, forming an ablation study to analyse the role of each component.

The setup is reiterated below in Table 4.12 to reduce flipping back and forth. The corresponding results of Experiments 1-6 are summarised in Table 4.13.

	Pretrain	Fine-tune	Train	Numeric	Regression
	MAE	MAE	LLaVA	Tokenisation	Head
Experiment 0			$\checkmark$		
Experiment 1		$\checkmark$	$\checkmark$		
Experiment 2	$\checkmark$		$\checkmark$		
Experiment 3	$\checkmark$	$\checkmark$	$\checkmark$		
<b>Experiment 4</b>	$\checkmark$	$\checkmark$			
Experiment 5		$\checkmark$			
Experiment 6	$\checkmark$				

Table 4.12 LLaVA-IntelliPrint Ablation Study on different training stages

#### Flow Rate Result Analysis

In Experiment 2, the model was pre-trained using a Masked Autoencoder without subsequent fine-tuning, resulting in a Mean Absolute Error of 45.78%. While this improvement over the baseline indicates that pre-training enables the model to learn useful data representations, the performance is still not optimal. This suggests that pre-training alone, without task-specific

Experiment Number	Mean Absolute Error	Consistency Percentage	Extrusion Quality Accuracy	Extrusion Quality Precision	Extrusion Quality Recall	Extrusion Quality F1	BERTScore Precision	BERTScore Recall	BERTScore F1
				[0.97	[0.21	[0.34			
0	62.4	1.0	0.51	0.2	0.13	0.17	0.91	0.91	0.91
				0.45]	0.96]	0.63]			
				[0.97	[0.90	[0.94			
1	13.86	0.97	0.89	0.56	0.90	0.64	0.92	0.92	0.92
				0.96]	0.86]	0.91]			
				[0.78	[0.80	[0.79			
2	45.78	0.95	0.63	0.2	0.56	0.29	0.91	0.91	0.91
				0.94]	0.45]	0.61]			
				[0.97	[0.97	[0.92			
3	13.8	0.96	0.9	0.50	0.81	0.02	0.91	0.91	0.91
				0.95]	0.98]	0.92]			
				[0	[0	[0			
4	N/A*	0	0	0	0	0	0.81	0.82	0.81
				0]	0]	0]			
				[0	[0	[0			
5	N/A*	0	0	0	0	0	0.81	0.81	0.81
				0]	0]	0]			
				0]	[0	0]			
6	N/A*	0	0	0	0	0	0.80	0.81	0.91
				01	01	01			

Table 4.13 Ablation study results for LLaVA-IntelliPrint, where N/A indicates the models failed to predict any numerical value for the flow rate

adaptation, is insufficient for fully capturing the complex relationship between the learned representations and the specific task of predicting flow rate.

Experiment 1, which focused on fine-tuning the model without prior pre-training, resulted in a significantly lower MAE of 13.86 %. This highlights the critical role of fine-tuning, as it allows the model to adapt specifically to the dataset at hand, better associating the input features with the flow rate.

Experiment 3, which combined both pre-training and fine-tuning, achieved the **best predictive accuracy with an MAE of 13.8%**. This demonstrates that while pre-training helps the model to learn general representations, targeted fine-tuning is essential to refine these representations for the specific task, thus significantly enhancing the model's ability to generalise and improve predictive accuracy.

Experiments 4, 5, and 6 assessed the performance of the LLaVA model when the MAE encoder was directly integrated without either pre-training or fine-tuning. In these cases, the model struggled to interpret the embeddings generated by the MAE encoder, leading to poor performance. The misalignment between the MAE-encoded features and the task of predicting flow rate, coupled with the lack of fine-tuning relevant to additive manufacturing, resulted in the model generating inaccurate outputs. Consequently, all three models failed to predict the flow rate and categorise extrusion quality effectively.

When comparing the baseline (Experiment 0) with Experiment 2, it is evident that pre-training with the MAE enhances representation learning, thereby improving model performance. However, the inclusion of fine-tuning, as demonstrated in Experiments 1 and 3, is crucial for aligning the learned representations with the flow rate prediction task, as evidenced by the significantly improved results.

Overall, Experiment 3 yielded the lowest mean absolute error of 13.8%, significantly outperforming the LLaVA baseline of 62.5%, indicating the effectiveness of our 4-stage training approach that both MAE pre-training and fine-tuning are critical. Especially fine-tuning, which allows the model to effectively associate input features with the flow rate.

#### **Classification Result Analysis**

The fine-tuned models in Experiments 1 and 3 achieved high accuracy rates of 89% and 90% in classifying extrusion quality (i.e. Under/good/over extrusion), outperforming the classification accuracy of 87.1% reported by Brion and Pattinson (2022b). However, it is noted that the classification recall, precision, and F1 scores for the "good-extrusion" class were lower than those for other classes. This discrepancy could be attributed to data imbalance, as the "good extrusion" label constitutes only about 10.04% of the original dataset. This bias imposes inherent difficulty in classifying underrepresented classes.

#### **Consistency Result Analysis**

The consistency of predictions was high, with results showing over 95% consistency. This can be attributed to the nature of causal inference in language models, where later tokens are influenced by previous predictions, thereby enforcing consistent prediction outcomes for both flow rate and classification tasks.

#### **Text Generation Quality Analysis**

The BERTScores provide an indication of the semantic quality of the generated text, with BERTScore (Precision, Recall, F1) values ranging from 0 to 1. A score of 0 represents no semantic similarity, while a score of 1 indicates a perfect semantic match between the candidate and reference texts. The higher score of BERTScores above 0.9 in Experiments 1-3 suggests that the generated text is coherent and of high quality, closely aligning with the reference text. In contrast, for Experiments 4-6, where the ViT and LLaMA models were not properly aligned, the BERTScore dropped to approximately 0.8, indicating a decline in semantic quality.

#### **Qualitative Analysis**

In addition to quantitative analysis, qualitative analysis was conducted to explore potential reasons for errors. Three representative examples were selected based on common error patterns identified:

#### 1. Consistency Error Due to Ambiguous Extrusion Categorisation Label

- Image: Figure 4.5
- Prompt: Estimate the flow rate and provide a detailed description of the image.
- **Reference**: Examining closely, this seems to be a material extrusion process, possibly additive manufacturing, with focus on the extrusion nozzle.Upon inspection, the extrusion nozzle demonstrates a flow rate of 112 %, ensuring consistent material flow and layer adhesion. This is essential for producing functional prototypes and end-use parts with precise geometries.
- **Prediction**: The focal point of this image is the extrusion nozzle, which suggests the operation of a material extrusion process, possibly within additive manufacturing. The flow rate through this extrusion nozzle appears optimal, suggesting favourable extrusion conditions with a flow rate of 112 %. This steady flow promotes uniform material distribution, which is vital for intricate designs and complex geometries.
- Analysis: This case was labelled as a consistency error due to the prediction of a flow rate of 112%, which was incorrectly categorised as "favourable" extrusion. The issue arises from a misalignment in the dataset's labelling, where a flow rate of 112% should have been categorised as over-extrusion rather than favourable. There is ambiguity in extrusion quality categorisation, especially on what flow rate should be classified as "good extrusion". Therefore, even though our prediction closely matches the reference, it is still marked as inconsistent, indicating a need for improved data labelling.

#### 2. Misclassification of Desired Shape and Erroneous Flow Rate Prediction

- Image: Figure 4.6
- **Prompt:** Estimate the flow rate and describe the image in detail.
- **Reference**: The image highlights the extrusion nozzle, a key element in the Fused Filament Fabrication (FFF) additive manufacturing process. There are clear indications of over-extrusion from this extrusion nozzle, potentially functioning

at 162 % flow rate. Over-extrusion can occur due to a high feed rate or excessive temperature, leading to problems like nozzle clogging or inadequate surface finish.

- **Prediction**: The focal point of this image is the extrusion nozzle, which suggests the operation of a material extrusion process, possibly within additive manufacturing. The extrusion nozzle exhibits signs of under-extrusion, implying a flow rate of 42 %. This diminished flow rate may stem from issues such as improper filament tension, insufficient cooling, or even variations in environmental temperature, all of which affect material flow dynamics during the printing process.
- Analysis: This error, with a high MAE of 120%, likely arose because the model interpreted the layer-by-layer pattern as part of the intended geometry rather than a consequence of over-extrusion. The focus on the nozzle area may have misled the model into identifying signs of under-extrusion. This issue has been observed across multiple examples, leading to significant MAEs.

#### 3. Erroneous Estimation Due to Blurry Image

- Image: Figure 4.7
- **Prompt:** Estimate the flow rate and provide a thorough analysis of the image.
- **Reference**: Let's examine the nozzle of this 3D printer. This extrusion nozzle exhibits signs of under-extrusion, suggesting that the flow rate may be 69 %. Under-extrusion commonly occurs when the feed rate is too low or the nozzle temperature is insufficient, resulting in inadequate material deposition.
- **Prediction**: The focal point of this image is the extrusion nozzle, which suggests the operation of a material extrusion process, possibly within additive manufacturing. Upon inspecting the extrusion nozzle, it becomes apparent that the flow rate is 188 %. This higher-than-ideal flow rate suggests over-extrusion, potentially causing issues such as poor surface finish and dimensional inaccuracies.
- Analysis: The high discrepancy between the predicted and reference flow rates could be attributed to the blurred and out-of-focus nature of the image. The model may have misinterpreted the filament's appearance, leading to an incorrect categorisation as over-extrusion. This issue is prevalent in cases where the image quality is poor, making accurate estimation challenging.





Fig. 4.6 Example 2 Image



Fig. 4.7 Example 3 Image

#### Summary of Section 4.3.3

Overall, LLaVA-IntelliPrint has demonstrated impressive capabilities in predicting flow rates, explaining the rationale behind predictions, and providing suggestions to address identified issues. However, potential improvements could be made to the dataset to ensure image clarity and consistency in categorising under-extrusion, good extrusion, and over-extrusion. Additionally, the model's limitations in distinguishing between intended patterns and extrusion failures could be further investigated. Future work should focus on enhancing the dataset and refining the model's ability to accurately interpret complex 3D printing scenarios.

# 4.3.4 LLaVA-IntelliPrint Ablation Study: Experiment 7-8 Numeric Tokenisation

In addition to the ablation studies, further experiments were conducted to explore techniques that could potentially enhance the model's performance. In large language models, text generation occurs token by token, with each subsequent token influenced by the previous tokens in the sequence due to the causal inference nature of these models. For instance, LLaMA 2 has a vocabulary of 30,000 tokens, with numerical tokens represented only by the digits 0-9. Consequently, any number greater than 9 is represented by multiple tokens (e.g., the number 284 is represented by three tokens: "2", "8", "4" rather than a single token "284"). We hypothesised that predicting numerical values as whole tokens, instead of token by token, might improve numerical estimation accuracy.

Due to computational constraints, this experiment was conducted using the setup from Experiment 3 (our best-performing model) and Experiment 0 (Baseline). The effect of numerical tokenisation on Experiment 3 is reported as Experiment 7, while its effect on Experiment 0 is reported as Experiment 8.

The experimental setup is summarised in Table 4.14, and the results are presented in Table 4.15.

	Pretrain MAE	Fine-tune MAE	Train LLaVA	Numeric Tokenisation	Regression Head
Experiment 7	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	
Experiment 8			$\checkmark$	$\checkmark$	

Table 4.14 LLaVA-IntelliPrint Numeric Tokenisation Ablation Study

Experiment Number	Mean Absolute Error	Consistency Percentage	Extrusion Quality Accuracy	Extrusion Quality Precision	Extrusion Quality Recall	Extrusion Quality F1	BERTScore Precision	BERTScore Recall	BERTScore F1
7	14.29	0.95	0.90	[0.97 0.61 0.91]	[0.91 0.68 0.96]	[0.94 0.64 0.94]	0.91	0.92	0.91
8	63.56	0.96	0.48	[0.96 0.18 0.42]	[0.56 0.20 0.92]	[0.35 0.21 0.61]	0.91	0.92	0.91

Table 4.15 Tokenisation Experiment Results

When compared Experiment 3 with 7, 0 with 8, there is no clear benefit from combining numerical tokens into whole units. In fact, the MAE increased from 13.80% to 14.29% and 62.4% to 63.56%, indicating worse performance. This may be due to the fact that treating numerical tokens as whole units can make the cross-entropy loss function less effective in estimating flow rate. For example, if the reference value is 13 and the prediction is 18, treating numbers separately allows for some alignment, as the first digit "1" matches and the loss is calculated on the differences between "3" and "8". However, if numbers are treated as whole units, the loss of "13" to "18" is treated similarly to any two-digit number (e.g. "13" to "99"), potentially weakening the loss function's ability to make accurate numerical predictions.

This analysis underscores the need for careful consideration when altering tokenisation strategies in language models, particularly when dealing with numerical data. The find-ings indicate that the conventional token-by-token approach may be more robust for tasks involving numerical prediction.

## 4.3.5 LLaVA-IntelliPrint Ablation Study: Experiment 9-10 Regression Head

This ablation study aims to examine the effectiveness of the novel architecture of adding a regression head during Vision-Language Model (VLM) training. Due to computational

limitations, this study was conducted only on the baseline model (Experiment 0) and the best-performing model (Experiment 3). The experimental setup is summarised in Table 4.16, and the results are presented in Table 4.17.

	Pretrain	Fine-tune	Train	Numeric	Regression
	MAE	MAE	LLaVA	Tokenisation	Head
Experiment 9	$\checkmark$	$\checkmark$	$\checkmark$		$\checkmark$
<b>Experiment 10</b>			$\checkmark$		$\checkmark$

Experiment Number	Mean Absolute Error	Consistency Percentage	Extrusion Quality Accuracy	Extrusion Quality Precision	Extrusion Quality Recall	Extrusion Quality F1	BERTScore Precision	BERTScore Recall	BERTScore F1
9	12.02	0.94	0.90	[0.98 0.58 0.93]	[0.88 0.82 0.95]	[0.93 0.68 0.94]	0.91	0.91	0.91
10	57 73	0.91	0.255	[0.44	[0.01	[0.03	0.01	0.90	0.01

Table 4.16 LLaVA-IntelliPrint Regression Head Ablation Study

0.36]

0.51]

0.91]

The results clearly demonstrate that the addition of a regression head significantly improves the model's regression performance. For the best-performing model (Experiment 3), the mean absolute error (MAE) decreased from 13.8 to 12.02, representing a 13% reduction. Similarly, for the baseline model (Experiment 0), the MAE decreased from 62.4 to 57.73, indicating a 7.5% reduction.

This improvement in regression accuracy, while maintaining language quality, classification accuracy, and consistency, validates the hypothesis that incorporating a joint Mean Squared Error (MSE) loss function alongside cross-entropy during training can enhance the overall regression performance of language models without undermining their natural language processing capabilities. The ability of the regression head to improve numerical prediction accuracy within a VLM framework establishes a new standard for integrating regression tasks in Large Language Models, particularly in specialised domains like 3D printing.

#### 4.3.6 Data Efficiency Optimisation

During the training phase, rapid loss convergence was observed, leading to the hypothesis that a smaller training dataset could yield similar results. Considering that acquiring labelled free-text descriptions in the 3D printing domain is particularly challenging, a data efficiency

analysis was conducted to potentially broaden the applicability of the LLaVA-IntelliPrint framework across different domains.

In this experiment, a checkpoint was saved for every 10% increment in the training data (i.e., a saving step of 40K) for the setups described in Experiments 1, 2, and 3. These configurations were chosen as they produced meaningful and concise predictions. The results of these experiments are depicted in Figures 4.8, 4.9, and 4.10.



Fig. 4.8 Experiment 1: Number of Data Points vs Evaluation Metrics



Fig. 4.9 Experiment 2: Number of Data Points vs Evaluation Metrics



Fig. 4.10 Experiment 3: Number of Data Points vs Evaluation Metrics

The graphs indicate that the evaluation metrics remain relatively stable across varying amounts of training data, including flow rate MAE, classification accuracy, consistency percentage, and BERTScore F1. The standard deviation for these metrics, as shown in Table 4.18, further supports this observation.

	Mean Absolute	Consistency	Classification	BERTScore	
	Error std	Percentage std	Accuracy std	F1 std	
Experiment 1	0.64	0.005	0.005	0.005	
Experiment 2	0.81	0.002	0.005	0.005	
Experiment 3	0.53	0.005	0.005	0.005	

Table 4.18 Standard deviation of evaluation metrics across different experiments

The low standard deviation values indicate that the model had already converged within 10% of the original dataset (40K), suggesting that the entire 400K training dataset may not be necessary. The negligible performance improvements observed after training on 40K data points could be attributed to the dataset's homogeneity (i.e., the nozzle images from the 3D printing process share similar distributions). Once the model fully learns the underlying patterns, additional data does not provide significant benefits. For this specific task, a substantially smaller dataset can achieve performance comparable to that obtained with a much larger dataset. This insight is particularly valuable in fields where acquiring large labelled datasets is challenging, enhancing the versatility and adaptability of the LLaVA-IntelliPrint framework to other specialised applications.

## 4.4 Chapter 4 Summary

Chapter 4 first validated the effectiveness of the MAE Vision Encoder and then validated the LLaVA-IntelliPrint model in enhancing additive manufacturing processes. The Masked AutoEncoder demonstrated robust performance in reconstructing 3D printing images and accurately estimating flow rates, highlighting the importance of the fine-tuning step in aligning image features with flow rate estimation. The LLaVA-IntelliPrint model achieved impressive results in flow rate estimation (12.02% Mean Absolute Error), extrusion quality classification (90%), language consistency (94%), and text generation quality, as evidenced by a BERTScore F1 of 0.91. This model has shown significant potential in enhancing human-computer interaction by autonomously detecting printing errors and providing corrective suggestions in real time.

Further enhancements revealed that only 10% (40K) of the original training dataset is sufficient for the model to converge, demonstrating its versatility and data efficiency. Additionally, the proposed regression task framework introduces a novel architecture that achieves state-of-the-art results in Vision-Language Model (VLM) regression tasks, marking a first in the existing literature. This approach has high potential for application in other fields with similar objectives.

## **Chapter 5**

## **Conclusions and Future Work**

This dissertation introduced and validated LLaVA-IntelliPrint, a novel framework combining Vision-Language Models (VLMs) with advanced vision transformers to enhance additive manufacturing processes. As the first of its kind in the industry, LLaVA-IntelliPrint demonstrates the efficacy of VLMs in accurately describing the quantitative properties of Fused Deposition Modelling (FDM) processes, showcasing their potential as embodied manufacturing agents.

The LLaVA-IntelliPrint model achieved state-of-the-art performance in key areas: flow rate estimation (12.02% Mean Absolute Error), extrusion quality classification (90% accuracy), language consistency (94%), and text generation quality (BERTScore F1 of 0.91). These results underscore the model's potential to enhance human-computer interaction by enabling real-time identification and correction of printing errors. Section 4.3.1 provides a detailed example of the model's performance.

Given its strong performance, LLaVA-IntelliPrint is highly relevant to practitioners in FDM. Its ability to accurately estimate flow rates and classify extrusion quality allows seamless integration into existing workflows, enhancing both process reliability and product quality. By diagnosing over-extrusion or under-extrusion from nozzle images, the model provides real-time feedback and corrective suggestions, significantly reducing time and material waste. The generated text not only identifies the causes of printing errors but also offers actionable solutions, effectively automating the error detection and correction process.

By automating the detection and correction of common FDM issues, LLaVA-IntelliPrint reduces the need for continuous human oversight, lowering operational costs and improving

scalability. This capability is particularly valuable in high-volume manufacturing, where consistent print quality is crucial.

Furthermore, the study revealed that the model could achieve comparable performance with only 40K training data points, demonstrating its data efficiency and versatility. The introduction of a regression head for LLMs and VLMs offers a novel approach to regression tasks, with potential benefits extending beyond manufacturing to broader machine-learning fields involving similar tasks.

In conclusion, LLaVA-IntelliPrint represents a significant advancement in integrating artificial intelligence into additive manufacturing. It provides a robust solution for real-time quality assurance and sets a new standard for intelligent manufacturing systems.

## 5.1 Future Work

We acknowledge the limitations of this work, which were influenced by time constraints (4 months) and computational resources (1000 GPU hours). Based on these limitations, we propose the following areas for future research.

### 5.1.1 Upgrade the Language Model

During the course of this project, Meta AI released their latest version of LLaMA 3 (AI@Meta, 2024), which has been described as the "most capable openly available LLM to date". In this work, we utilised LLaMA 2 as the language model within the LLaVA framework. Future work could explore upgrading to LLaMA 3 to determine if a more powerful language model can further enhance the performance of LLaVA-IntelliPrint, potentially leading to more accurate flow rate predictions that matches the performance of a pure vision encoder.

## 5.1.2 Post-processing Vision Encoder and Language Model for Enhanced Flow Rate Estimation

As suggested in Section 4, the vision encoder achieved a mean absolute error of 6.49% in flow rate estimation, whereas the combined Vision-Language Model (VLM) approach resulted in an MAE of 12.02%. Until we identify methods to further improve the VLM regression performance, one approach could involve a post-processing step where the flow rate tokens predicted by the VLM are replaced with the values estimated by the vision encoder. By combining the numerical accuracy of the vision encoder with the detailed

process descriptions generated by the language model, we can potentially enhance both the accuracy and interpretability of the flow rate predictions.

### 5.1.3 Rigorous Testing of the LLM Regression Head

Due to computational limitations, we only tested the novel architecture of adding a regression head to the language model (LLM) on the best-performing model and the baseline LLaVA model. Future work should involve more extensive testing across a variety of models and datasets to thoroughly evaluate the architecture's capability for regression tasks in LLMs. This would help establish the generalisability and robustness of this approach.

## 5.1.4 Refining Extrusion Quality Labels

As discussed in Section 4.3.3, the current extrusion quality labels are ambiguous, lacking a clear categorisation that aligns with specific flow rates. For instance, a flow rate of 112% is labelled as "good extrusion", which could also be interpreted as over-extrusion. This ambiguity undermines the model's classification performance and could be addressed by filtering and refining the dataset to establish more precise and consistent labels.

## 5.1.5 Expanding the Dataset with a Greater Variety of Information

Expanding the current 3D printing dataset with additional labelled data could significantly enhance LLaVA-IntelliPrint's capabilities for various tasks. For example, including **G-code**, the language used to control 3D printers, in the training data could enable the model to directly output commands that correct deviations in flow rate. Additionally, incorporating nozzle and bed **temperature** data, which is a crucial parameter in 3D printing, could further improve the model's ability to assess and optimise print quality. This expansion could make the model more versatile and effective in addressing a broader range of challenges in additive manufacturing.

## 5.1.6 Evaluating the Model on Different Datasets

The current model was trained and tested on a dataset with similar images and labels. It would be valuable to explore the model's performance on other additive manufacturing datasets with varying characteristics. This would provide insights into the model's robustness and generalisability, ensuring its applicability across different contexts and potentially leading to improvements in its architecture and training process.

## References

- 1. Abacha, A. B., Sarrouti, M., Demner-Fushman, D., Hasan, S. A., & Müller, H. (n.d.). Overview of the VQA-Med Task at ImageCLEF 2021: Visual Question Answering and Generation in the Medical Domain.
- Agrawal, A., Batra, D., Parikh, D., & Kembhavi, A. (2018, June). Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering [arXiv:1712.00377 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1712.00377
- Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C. L., Batra, D., & Parikh, D. (2016, October). VQA: Visual Question Answering [arXiv:1505.00468 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1505.00468
- 4. AI@Meta. (2024). Llama 3 Model Card. https://github.com/meta-llama/llama3/blob/ main/MODEL\_CARD.md
- Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., ... Simonyan, K. (2022, November). Flamingo: A Visual Language Model for Few-Shot Learning [arXiv:2204.14198 [cs]]. Retrieved April 7, 2024, from http://arxiv.org/abs/2204.14198
- Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018, March). Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering [arXiv:1707.07998 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1707.07998
- Aoyagi, K., Wang, H., Sudo, H., & Chiba, A. (2019). Simple method to construct process maps for additive manufacturing using a support vector machine. *Additive Manufacturing*, 27, 353–362. https://doi.org/10.1016/j.addma.2019.03.013
- Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., & Zhou, J. (2023, October). Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond [arXiv:2308.12966 [cs]]. Retrieved August 14, 2024, from http://arxiv.org/abs/2308.12966
- Banerjee, S., & Lavie, A. (2005, June). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In J. Goldstein, A. Lavie, C.-Y. Lin, & C. Voss (Eds.), *Proceedings of the ACL Workshop on Intrinsic* and Extrinsic Evaluation Measures for Machine Translation and/or Summarization (pp. 65–72). Association for Computational Linguistics. Retrieved August 3, 2024, from https://aclanthology.org/W05-0909

- Bao, H., Dong, L., Piao, S., & Wei, F. (2022, September). BEiT: BERT Pre-Training of Image Transformers [arXiv:2106.08254 [cs]]. Retrieved April 7, 2024, from http: //arxiv.org/abs/2106.08254
- Brion, D. A. J., & Pattinson, S. W. (2022a). Quantitative and Real-Time Control of 3D Printing Material Flow Through Deep Learning. *Advanced Intelligent Systems*, 4(11), 2200153. https://doi.org/10.1002/aisy.202200153
- 12. Brion, D. A. J., & Pattinson, S. W. (2022b). Generalisable 3D printing error detection and correction via multi-head neural networks. *Nature Communications*, *13*(1), 4654. https://doi.org/10.1038/s41467-022-31985-y
- Brion, D. A., Shen, M., & Pattinson, S. W. (2022). Automated recognition and correction of warp deformation in extrusion additive manufacturing. *Additive Manufacturing*, 56, 102838. https://doi.org/10.1016/j.addma.2022.102838
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020, July). Language Models are Few-Shot Learners [arXiv:2005.14165 [cs]]. Retrieved April 7, 2024, from http://arxiv.org/abs/2005.14165
- Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021, May). Emerging Properties in Self-Supervised Vision Transformers [arXiv:2104.14294 [cs]]. Retrieved July 27, 2024, from http://arxiv.org/abs/2104.14294
- 16. Chen, C.-T., & Gu, G. X. (2019). Machine learning for composite materials. *MRS Communications*, 9(2), 556–566. https://doi.org/10.1557/mrc.2019.32
- 17. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Dhariwal, P., Luan, D., & Sutskever, I. (2020). Generative Pretraining from Pixels.
- Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A. J., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., Kolesnikov, A., Puigcerver, J., Ding, N., Rong, K., Akbari, H., Mishra, G., Xue, L., Thapliyal, A., Bradbury, J., ... Soricut, R. (2023, June). PaLI: A Jointly-Scaled Multilingual Language-Image Model [arXiv:2209.06794 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/abs/2209.06794
- Chen, X., Xie, S., & He, K. (2021, August). An Empirical Study of Training Self-Supervised Vision Transformers [arXiv:2104.02057 [cs]]. Retrieved July 28, 2024, from http://arxiv.org/abs/2104.02057
- Chen, Y.-C., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., & Liu, J. (2020, July). UNITER: UNiversal Image-TExt Representation Learning [arXiv:1909.11740 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/abs/1909.11740
- Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., & Xing, E. P. (2023, March). Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%\* ChatGPT Quality | LMSYS Org. Retrieved July 31, 2024, from https://lmsys.org/blog/2023-03-30-vicuna

- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., ... Fiedel, N. (2022, October). PaLM: Scaling Language Modeling with Pathways [arXiv:2204.02311 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/abs/2204.02311
- Cordonnier, J.-B., Loukas, A., & Jaggi, M. (2020, January). On the Relationship between Self-Attention and Convolutional Layers [arXiv:1911.03584 [cs, stat]]. Retrieved July 27, 2024, from http://arxiv.org/abs/1911.03584
- Cunha, F. G., Santos, T. G., & Xavier, J. (2021). In Situ Monitoring of Additive Manufacturing Using Digital Image Correlation: A Review. *Materials*, 14(6), 1511. https://doi.org/10.3390/ma14061511
- Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., & Hoi, S. (2023, June). InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning [arXiv:2305.06500 [cs]]. Retrieved August 14, 2024, from http://arxiv.org/abs/2305.06500
- Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J. M. F., Parikh, D., & Batra, D. (2017, August). Visual Dialog [arXiv:1611.08669 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1611.08669
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [arXiv:1810.04805 [cs]]. Retrieved May 31, 2024, from http://arxiv.org/abs/1810.04805
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021, June). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [arXiv:2010.11929 [cs]]. Retrieved April 7, 2024, from http://arxiv.org/abs/2010.11929
- Equbal, A., Sood, A. K., & Mahapatra, S. (2011). Prediction of dimensional accuracy in fused deposition modelling: A fuzzy logic approach [Publisher: Inderscience Publishers]. *International Journal of Productivity and Quality Management*, 7(1), 22–43. https://doi.org/10.1504/IJPQM.2011.03773
- Estelle, K. T., & Gozen, B. A. (2024). Precision flow rate control during micro-scale material extrusion by iterative learning of pressure-flow rate relationships. *Additive Manufacturing*, 82, 104031. https://doi.org/10.1016/j.addma.2024.104031
- Farhan Khan, M., Alam, A., Ateeb Siddiqui, M., Saad Alam, M., Rafat, Y., Salik, N., & Al-Saidan, I. (2021). Real-time defect detection in 3D printing using machine learning. *Materials Today: Proceedings*, 42, 521–528. https://doi.org/10.1016/j.matpr.2020.10. 482
- 32. Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., & Rohrbach, M. (2016, September). Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding [arXiv:1606.01847 [cs]]. Retrieved July 29, 2024, from http: //arxiv.org/abs/1606.01847

- Gan, Z., Li, L., Li, C., Wang, L., Liu, Z., & Gao, J. (2022, October). Vision-Language Pre-training: Basics, Recent Advances, and Future Trends [arXiv:2210.09263 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2210.09263
- Gao, W., Zhang, Y., Ramanujan, D., Ramani, K., Chen, Y., Williams, C. B., Wang, C. C., Shin, Y. C., Zhang, S., & Zavattieri, P. D. (2015). The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design*, 69, 65–89. https://doi.org/10.1016/j.cad.2015.04.001
- Goh, G. D., Sing, S. L., & Yeong, W. Y. (2021). A review on machine learning in 3D printing: Applications, potential, and challenges. *Artificial Intelligence Review*, 54(1), 63–94. https://doi.org/10.1007/s10462-020-09876-9
- Goodfellow, I., Bengio, Y., & Courville, A. (2017, January). *Deep Learning*. MIT Press.
- Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., & Parikh, D. (2017, May). Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering [arXiv:1612.00837 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/ abs/1612.00837
- Gu, S., Choi, M., Park, H., Jeong, S., Doh, J., & Park, S.-i. (2023). Application of artificial intelligence in additive manufacturing. *JMST Advances*, 5(4), 93–104. https://doi.org/10.1007/s42791-023-00057-7
- Guo, D., Xu, C., & Tao, D. (2019, February). Image-Question-Answer Synergistic Network for Visual Dialog [arXiv:1902.09774 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1902.09774
- 40. Guo, J., Wu, J., Sun, Z., Long, J., & Zhang, S. (2019). Fault Diagnosis of Delta 3D Printers Using Transfer Support Vector Machine With Attitude Signals [Conference Name: IEEE Access]. *IEEE Access*, 7, 40359–40368. https://doi.org/10.1109/ACCESS. 2019.2905264
- 41. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2021, December). Masked Autoencoders Are Scalable Vision Learners [arXiv:2111.06377 [cs]]. Retrieved June 25, 2024, from http://arxiv.org/abs/2111.06377
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90
- He, K., Zhang, Q., & Hong, Y. (2019). Profile monitoring based quality control method for fused deposition modeling process. *Journal of Intelligent Manufacturing*, 30(2), 947–958. https://doi.org/10.1007/s10845-018-1424-9
- 44. Hinck, M., Olson, M. L., Cobbley, D., Tseng, S.-Y., & Lal, V. (2024, June). LLaVA-Gemma: Accelerating Multimodal Foundation Models with a Compact Language Model [arXiv:2404.01331 [cs]]. Retrieved August 4, 2024, from http://arxiv.org/abs/2404.01331

- 45. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. v. d., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., ... Sifre, L. (2022, March). Training Compute-Optimal Large Language Models [arXiv:2203.15556 [cs]]. Retrieved April 7, 2024, from http://arxiv.org/abs/ 2203.15556
- 46. Holzmond, O., & Li, X. (2017). In situ real time defect detection of 3D printed parts. *Additive Manufacturing*, *17*, 135–142. https://doi.org/10.1016/j.addma.2017.08.003
- 47. Hou, L., Samaras, D., Kurc, T., Gao, Y., & Saltz, J. (2017). ConvNets with Smooth Adaptive Activation Functions for Regression [ISSN: 2640-3498]. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 430–439. Retrieved August 3, 2024, from https://proceedings.mlr.press/v54/hou17a.html
- 48. Huang, T., Wang, S., Yang, S., & Dai, W. (2021). Statistical process monitoring in a specified period for the image data of fused deposition modeling parts with consistent layers. *Journal of Intelligent Manufacturing*, *32*(8), 2181–2196. https://doi.org/10.1007/s10845-020-01628-4
- Hudson, D. A., & Manning, C. D. (2019, May). GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering [arXiv:1902.09506 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1902.09506
- Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q. V., Sung, Y., Li, Z., & Duerig, T. (2021, June). Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision [arXiv:2102.05918 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2102.05918
- Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. I., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., & Sayed, W. E. (2023, October). Mistral 7B [arXiv:2310.06825 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/ 2310.06825
- Jiang, J., Xiong, Y., Zhang, Z., & Rosen, D. W. (2022). Machine learning integrated design for additive manufacturing. *Journal of Intelligent Manufacturing*, 33(4), 1073– 1086. https://doi.org/10.1007/s10845-020-01715-6
- 53. Jin, Z., Zhang, Z., & Gu, G. X. (2020). Automated Real-Time Detection and Prediction of Interlayer Imperfections in Additive Manufacturing Processes Using Artificial Intelligence [\_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.201900130]. Advanced Intelligent Systems, 2(1), 1900130. https://doi.org/10.1002/aisy.201900130
- Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., & Girshick, R. (2016, December). CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning [arXiv:1612.06890 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1612.06890
- 55. Kingma, D. P., & Ba, J. (2017, January). Adam: A Method for Stochastic Optimization [arXiv:1412.6980 [cs]]. Retrieved August 6, 2024, from http://arxiv.org/abs/1412.6980

- Kuhnle, A., & Copestake, A. (2017, April). ShapeWorld A new test methodology for multimodal language understanding [arXiv:1704.04517 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1704.04517
- Kuprashevich, M., & Tolstykh, I. (2023, September). MiVOLO: Multi-input Transformer for Age and Gender Estimation [arXiv:2307.04616 [cs]]. Retrieved August 2, 2024, from http://arxiv.org/abs/2307.04616
- Li, C., Gan, Z., Yang, Z., Yang, J., Li, L., Wang, L., & Gao, J. (2023, September). Multimodal Foundation Models: From Specialists to General-Purpose Assistants [arXiv:2309.10020 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2309.10020
- Li, C., Liu, H., Li, L. H., Zhang, P., Aneja, J., Yang, J., Jin, P., Hu, H., Liu, Z., Lee, Y. J., & Gao, J. (2022, October). ELEVATER: A Benchmark and Toolkit for Evaluating Language-Augmented Visual Models [arXiv:2204.08790 [cs]]. Retrieved May 13, 2024, from http://arxiv.org/abs/2204.08790
- Li, C., Wong, C., Zhang, S., Usuyama, N., Liu, H., Yang, J., Naumann, T., Poon, H., & Gao, J. (2023, June). LLaVA-Med: Training a Large Language-and-Vision Assistant for Biomedicine in One Day [arXiv:2306.00890 [cs]]. Retrieved May 24, 2024, from http://arxiv.org/abs/2306.00890
- Li, J., Li, D., Savarese, S., & Hoi, S. (2023, June). BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models [arXiv:2301.12597 [cs]]. Retrieved July 23, 2024, from http://arxiv.org/abs/2301.12597
- Li, J., Li, D., Xiong, C., & Hoi, S. (2022, February). BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation [arXiv:2201.12086 [cs]]. Retrieved May 16, 2024, from http://arxiv.org/abs/2201.12086
- Li, J., Selvaraju, R. R., Gotmare, A. D., Joty, S., Xiong, C., & Hoi, S. (2021, October). Align before Fuse: Vision and Language Representation Learning with Momentum Distillation [arXiv:2107.07651 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/ abs/2107.07651
- Li, L. H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., Chang, K.-W., & Gao, J. (2022, June). Grounded Language-Image Pre-training [arXiv:2112.03857 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/ abs/2112.03857
- Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., & Gao, J. (2020, July). Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks [arXiv:2004.06165 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/abs/2004.06165
- 66. Liu, H., Li, C., Li, Y., & Lee, Y. J. (2023, October). Improved Baselines with Visual Instruction Tuning [arXiv:2310.03744 [cs]]. Retrieved May 13, 2024, from http: //arxiv.org/abs/2310.03744
- Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023, December). Visual Instruction Tuning [arXiv:2304.08485 [cs]]. Retrieved May 13, 2024, from http://arxiv.org/abs/2304. 08485

- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021, August). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows [arXiv:2103.14030 [cs]]. Retrieved July 27, 2024, from http://arxiv.org/abs/2103.14030
- Lu, J., Batra, D., Parikh, D., & Lee, S. (2019, August). ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks [arXiv:1908.02265 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/abs/1908.02265
- Margadji, C., Brion, D. A., & Pattinson, S. W. (2024). Iterative learning for efficient additive mass production. *Additive Manufacturing*, 89, 104271. https://doi.org/10. 1016/j.addma.2024.104271
- 71. Marin, D., Chang, J.-H. R., Ranjan, A., Prabhu, A., Rastegari, M., & Tuzel, O. (2023). Token Pooling in Vision Transformers for Image Classification. 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 12–21. https://doi.org/10. 1109/WACV56688.2023.00010
- Marino, K., Rastegari, M., Farhadi, A., & Mottaghi, R. (2019, September). OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge [arXiv:1906.00067 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1906.00067
- Mbodj, N. G., Abuabiah, M., Plapper, P., El Kandaoui, M., & Yaacoubi, S. (2021). Bead Geometry Prediction in Laser-Wire Additive Manufacturing Process Using Machine Learning: Case of Study. *Applied Sciences*, 11(24), 11949. https://doi.org/10.3390/ app112411949
- 74. Ojala, R., & Seppänen, A. (2024). Lightweight Regression Model with Prediction Interval Estimation for Computer Vision-based Winter Road Surface Condition Monitoring. *IEEE Transactions on Intelligent Vehicles*, 1–13. https://doi.org/10.1109/TIV. 2024.3371104
- Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., ... Bojanowski, P. (2024, February). DINOv2: Learning Robust Visual Features without Supervision [arXiv:2304.07193 [cs]]. Retrieved August 2, 2024, from http://arxiv.org/abs/2304. 07193
- 76. Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002, July). Bleu: A Method for Automatic Evaluation of Machine Translation. In P. Isabelle, E. Charniak, & D. Lin (Eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 311–318). Association for Computational Linguistics. https://doi.org/ 10.3115/1073083.1073135
- 77. Pham, H., Dai, Z., Ghiasi, G., Kawaguchi, K., Liu, H., Yu, A. W., Yu, J., Chen, Y.-T., Luong, M.-T., Wu, Y., Tan, M., & Le, Q. V. (2023, April). Combined Scaling for Zero-shot Transfer Learning [arXiv:2111.10050 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2111.10050
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021, February). Learning Transferable Visual Models From Natural Language Supervision [arXiv:2103.00020 [cs]]. Retrieved April 4, 2024, from http://arxiv.org/abs/2103.00020

- 79. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2023, September). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer [arXiv:1910.10683 [cs, stat]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1910.10683
- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022, April). Hierarchical Text-Conditional Image Generation with CLIP Latents [arXiv:2204.06125 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2204.06125
- Rao, P. K., Liu, J. (, Roberson, D., & Kong, Z. ( (2015). Sensor-Based Online Process Fault Detection in Additive Manufacturing. *Volume 2: Materials; Biomanufacturing; Properties, Applications and Systems; Sustainable Manufacturing*, V002T04A010. https://doi.org/10.1115/MSEC2015-9389
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022, April). High-Resolution Image Synthesis with Latent Diffusion Models [arXiv:2112.10752 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2112.10752
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015, January). ImageNet Large Scale Visual Recognition Challenge [arXiv:1409.0575 [cs]]. Retrieved July 24, 2024, from http://arxiv.org/abs/1409.0575
- Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., & Norouzi, M. (2022, May). Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding [arXiv:2205.11487 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/ abs/2205.11487
- Santoro, A., Raposo, D., Barrett, D. G. T., Malinowski, M., Pascanu, R., Battaglia, P., & Lillicrap, T. (2017, June). A simple neural network module for relational reasoning [arXiv:1706.01427 [cs]]. Retrieved August 3, 2024, from http://arxiv.org/abs/1706. 01427
- Schwenk, D., Khandelwal, A., Clark, C., Marino, K., & Mottaghi, R. (2022). A-OKVQA: A Benchmark for Visual Question Answering Using World Knowledge [Series Title: Lecture Notes in Computer Science]. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), *Computer Vision ECCV 2022* (pp. 146–162, Vol. 13668). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-20074-8\_9
- Sharma, P., Ding, N., Goodman, S., & Soricut, R. (2018, July). Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In I. Gurevych & Y. Miyao (Eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2556–2565). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1238
- Shen, X., Yao, J., Wang, Y., & Yang, J. (2004). Density Prediction of Selective Laser Sintering Parts Based on Artificial Neural Network. In F.-L. Yin, J. Wang, & C. Guo (Eds.), *Advances in Neural Networks - ISNN 2004* (pp. 832–840). Springer. https: //doi.org/10.1007/978-3-540-28648-6\_133

- Shi, W., Hu, Z., Bin, Y., Liu, J., Yang, Y., Ng, S.-K., Bing, L., & Lee, R. K.-W. (2024, June). Math-LLaVA: Bootstrapping Mathematical Reasoning for Multimodal Large Language Models [arXiv:2406.17294 [cs]]. Retrieved August 4, 2024, from http://arxiv.org/abs/2406.17294
- Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., & Rohrbach, M. (2019, May). Towards VQA Models That Can Read [arXiv:1904.08920 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1904.08920
- Sood, A. K., Ohdar, R. K., & Mahapatra, S. S. (2012). Experimental investigation and empirical modelling of FDM process for compressive strength improvement. *Journal* of Advanced Research, 3(1), 81–90. https://doi.org/10.1016/j.jare.2011.05.001
- Straub, J. (2015). Initial Work on the Characterization of Additive Manufacturing (3D Printing) Using Software Image Analysis. *Machines*, 3(2), 55–71. https://doi.org/10. 3390/machines3020055
- Suzuki, A., Shiba, Y., Ibe, H., Takata, N., & Kobashi, M. (2022). Machine-learning assisted optimization of process parameters for controlling the microstructure in a laser powder bed fused WC/Co cemented carbide. *Additive Manufacturing*, 59, 103089. https://doi.org/10.1016/j.addma.2022.103089
- Tan, H., & Bansal, M. (2019, December). LXMERT: Learning Cross-Modality Encoder Representations from Transformers [arXiv:1908.07490 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/abs/1908.07490
- Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., & Fidler, S. (2016, September). MovieQA: Understanding Stories in Movies through Question-Answering [arXiv:1512.02902 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1512.02902
- 96. Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., & Li, L.-J. (2016). YFCC100M: The New Data in Multimedia Research [arXiv:1503.01817 [cs]]. *Communications of the ACM*, 59(2), 64–73. https://doi.org/10.1145/2812802
- 97. Torta, S., & Torta, J. (2019). *3D printing: An introduction* [OCLC: on1024256466]. Mercury Learning; Information.
- Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021, January). Training data-efficient image transformers & distillation through attention [arXiv:2012.12877 [cs]]. Retrieved April 7, 2024, from http://arxiv.org/abs/2012.12877
- Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., & Jégou, H. (2021, April). Going deeper with Image Transformers [arXiv:2103.17239 [cs]]. Retrieved August 3, 2024, from http://arxiv.org/abs/2103.17239
- 100. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., ... Scialom, T. (2023, July). Llama 2: Open Foundation and Fine-Tuned Chat Models [arXiv:2307.09288 [cs]]. Retrieved April 7, 2024, from http://arxiv.org/abs/2307.09288

- 101. Trott, A., Xiong, C., & Socher, R. (2018, March). Interpretable Counting for Visual Question Answering [arXiv:1712.08697 [cs]]. Retrieved August 3, 2024, from http: //arxiv.org/abs/1712.08697
- 102. Tsimpoukelli, M., Menick, J., Cabi, S., Eslami, S. M. A., Vinyals, O., & Hill, F. (2021, July). Multimodal Few-Shot Learning with Frozen Language Models [arXiv:2106.13884 [cs]]. Retrieved May 9, 2024, from http://arxiv.org/abs/2106.13884
- 103. Vacareanu, R., Negru, V.-A., Suciu, V., & Surdeanu, M. (2024, April). From Words to Numbers: Your Large Language Model Is Secretly A Capable Regressor When Given In-Context Examples [arXiv:2404.07544 [cs]]. Retrieved June 23, 2024, from http://arxiv.org/abs/2404.07544
- 104. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023, August). Attention Is All You Need [arXiv:1706.03762 [cs]]. Retrieved July 24, 2024, from http://arxiv.org/abs/1706.03762
- 105. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th international conference on Machine learning - ICML '08*, 1096–1103. https://doi.org/ 10.1145/1390156.1390294
- 106. Vosahlik, D., Cech, J., Hanis, T., Konopisky, A., Rurtle, T., Svancar, J., & Twardzik, T. (2021). Self-Supervised Learning of Camera-based Drivable Surface Friction. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2773–2780. https://doi.org/10.1109/ITSC48978.2021.9564894
- 107. Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., Liu, Z., Liu, C., & Wang, L. (2022, December). GIT: A Generative Image-to-text Transformer for Vision and Language [arXiv:2205.14100 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2205.14100
- 108. Wang, P., Wu, Q., Shen, C., Hengel, A. v. d., & Dick, A. (2017, August). FVQA: Fact-based Visual Question Answering [arXiv:1606.05433 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/1606.05433
- 109. Wang, R.-J., Li, J., Wang, F., Li, X., & Wu, Q. (2009). ANN model for the prediction of density in Selective Laser Sintering [Publisher: Inderscience Publishers]. *International Journal of Manufacturing Research*. Retrieved August 14, 2024, from https://www. inderscienceonline.com/doi/10.1504/IJMR.2009.026579
- 110. Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Mohammed, O. K., Singhal, S., Som, S., & Wei, F. (2022, August). Image as a Foreign Language: BEiT Pretraining for All Vision and Vision-Language Tasks [arXiv:2208.10442 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/abs/2208.10442
- 111. Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., & Cao, Y. (2022, May). SimVLM: Simple Visual Language Model Pretraining with Weak Supervision [arXiv:2108.10904 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/abs/2108.10904
- Wenger, L., Strauß, S., & Hubbuch, J. (2022). Automated and dynamic extrusion pressure adjustment based on real-time flow rate measurements for precise ink dispensing in 3D bioprinting. *Bioprinting*, 28, e00229. https://doi.org/10.1016/j.bprint.2022.e00229
- 113. Wielitzka, M., Dagen, M., & Ortmaier, T. (2018). Sensitivity-based Road Friction Estimation in Vehicle Dynamics using the Unscented Kalman Filter [ISSN: 2378-5861]. 2018 Annual American Control Conference (ACC), 2593–2598. https://doi.org/10. 23919/ACC.2018.8431259
- 114. Wong, K. V., & Hernandez, A. (2012). A Review of Additive Manufacturing. ISRN Mechanical Engineering, 2012, 1–10. https://doi.org/10.5402/2012/208760
- 115. Wu, H., Wang, Y., & Yu, Z. (2015). In situ monitoring of FDM machine condition via acoustic emission. *The International Journal of Advanced Manufacturing Technology*. https://doi.org/10.1007/s00170-015-7809-4
- 116. Xu, J., Pan, Y., Pan, X., Hoi, S., Yi, Z., & Xu, Z. (2021, January). RegNet: Self-Regulated Network for Image Classification [arXiv:2101.00590 [cs, eess]]. Retrieved July 24, 2024, from http://arxiv.org/abs/2101.00590
- 117. Yadav, D., Jain, R., Agrawal, H., Chattopadhyay, P., Singh, T., Jain, A., Singh, S. B., Lee, S., & Batra, D. (2019, February). EvalAI: Towards Better Evaluation Systems for AI Agents [arXiv:1902.03570 [cs]]. Retrieved August 3, 2024, from http://arxiv.org/ abs/1902.03570
- 118. Yang, J., Li, C., Zhang, P., Xiao, B., Liu, C., Yuan, L., & Gao, J. (2022, April). Unified Contrastive Learning in Image-Text-Label Space [arXiv:2204.03610 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2204.03610
- 119. Yang, Y., Zha, K., Chen, Y.-C., Wang, H., & Katabi, D. (2021, May). Delving into Deep Imbalanced Regression [arXiv:2102.09554 [cs]]. Retrieved August 3, 2024, from http://arxiv.org/abs/2102.09554
- 120. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., & Wu, Y. (2022, June). CoCa: Contrastive Captioners are Image-Text Foundation Models [arXiv:2205.01917 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2205.01917
- 121. Yuan, L., Hou, Q., Jiang, Z., Feng, J., & Yan, S. (2021, June). VOLO: Vision Outlooker for Visual Recognition [arXiv:2106.13112 [cs]]. Retrieved August 3, 2024, from http://arxiv.org/abs/2106.13112
- 122. Yuan, L., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., Liu, C., Liu, M., Liu, Z., Lu, Y., Shi, Y., Wang, L., Wang, J., Xiao, B., Xiao, Z., ... Zhang, P. (2021, November). Florence: A New Foundation Model for Computer Vision [arXiv:2111.11432 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2111.11432
- 123. Zhang, H., Li, F., Zou, X., Liu, S., Li, C., Gao, J., Yang, J., & Zhang, L. (2023, March). A Simple Framework for Open-Vocabulary Segmentation and Detection [arXiv:2303.08131 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2303.08131
- 124. Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., & Gao, J. (2021, March). VinVL: Revisiting Visual Representations in Vision-Language Models [arXiv:2101.00529 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/abs/2101.00529

- 125. Zhang, T., Li, X., Fei, H., Yuan, H., Wu, S., Ji, S., Loy, C. C., & Yan, S. (2024, June). OMG-LLaVA: Bridging Image-level, Object-level, Pixel-level Reasoning and Understanding [arXiv:2406.19389 [cs]]. Retrieved July 23, 2024, from http://arxiv.org/ abs/2406.19389
- 126. Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020, February). BERTScore: Evaluating Text Generation with BERT [arXiv:1904.09675 [cs]]. Retrieved July 21, 2024, from http://arxiv.org/abs/1904.09675
- 127. Zhang, Y., Hare, J., & Prügel-Bennett, A. (2018, February). Learning to Count Objects in Natural Images for Visual Question Answering [arXiv:1802.05766 [cs]]. Retrieved August 2, 2024, from http://arxiv.org/abs/1802.05766
- 128. Zhang, Z., Fidan, I., & Allen, M. (2020). Detection of Material Extrusion In-Process Failures via Deep Learning. *Inventions*, 5(3), 25. https://doi.org/10.3390/inventions5030025
- 129. Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L. H., Zhou, L., Dai, X., Yuan, L., Li, Y., & Gao, J. (2021, December). RegionCLIP: Region-based Language-Image Pretraining [arXiv:2112.09106 [cs]]. Retrieved July 29, 2024, from http://arxiv.org/abs/2112.09106
- Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2017, November). Random Erasing Data Augmentation [arXiv:1708.04896 [cs]]. Retrieved July 13, 2024, from http://arxiv.org/abs/1708.04896
- 131. Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J. J., & Gao, J. (2019, December). Unified Vision-Language Pre-Training for Image Captioning and VQA [arXiv:1909.11059 [cs]]. Retrieved July 30, 2024, from http://arxiv.org/abs/1909.11059