



Universiteit
Leiden
The Netherlands

Data Science & Artificial Intelligence

Analysis and Visualization of AI-generated Amigurumi Crochet Patterns

Catherine Smeyers

Supervisors:
Tessa Verhoef & Rob Saunders

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

01/07/2025

Abstract

This paper explores the creative capabilities of current large language models (LLMs) by generating amigurumi crochet patterns based on image prompts. Amigurumi are small stuffed creatures fabricated using a hook and yarn. By integrating OpenAI’s image generation model DALL-E 3 and language models GPT-4o and o1, a pipeline is created that starts off by generating an image of a crochet project, which is used to then derive corresponding crochet instructions. The study focuses on three main questions: how to evaluate the quality of AI-generated patterns without physical artifacts, whether LLMs exhibit an understanding of underlying crochet logic, and to what extent AI can generate original designs. A systematic scoring method was developed to compare features extracted from generated patterns to those in the source images, analyzing both structural components and color accuracy. Additionally, qualitative experiments assess originality and logical adaptability through fictional creatures and targeted pattern modifications. The results indicate that both models are indeed capable of generating plausible crochet patterns adhering to most specifications from the source images. The suggested crochet pattern generation pipeline was found to be overall successful.

Contents

1	Introduction	1
1.1	Evolution of LLMs	2
1.2	Defining creativity	3
1.3	Thesis overview	4
2	Related Work	5
3	Methodology	8
3.1	Deciding on a pattern generation process	8
3.2	AI models	9
3.3	Feature extraction	10
3.4	Score calculation	12
4	Experiments	14
4.1	Data Collection of AI-generated crochet patterns	14
4.1.1	Collected Data and Results	14
4.1.2	Significance Test Model score averages	19
4.2	Visualization	19
4.2.1	2D Visualization	20
4.2.2	3D Visualization	23
4.2.3	Other visualization considerations	24
4.3	Exploring the originality of AI in Amigurumi pattern design	25
4.3.1	Mismatch image experiment	25
4.3.2	Mythical creatures experiment	27
4.4	Testing AI’s ability to adapt existing patterns	31
5	Conclusions	34
5.1	Results	34
5.2	Limitations	35
5.3	Future work	35
	References	38
6	Appendix	38

1 Introduction

There is a strong historical link between textile crafts, e.g., crocheting or knitting, and programming. A notable example is the Jacquard loom, introduced in 1801, which used punched cards to control the weaving of complex patterns. This mechanical innovation directly influenced Charles Babbage’s design of the Analytical Engine. The Analytical Engine is widely regarded as the primary design leading to the evolution of modern-day computers [Lan19]. Ada Lovelace, often considered the first computer programmer, famously drew a parallel between fabric arts and computational processes, noting that “the Analytical Engine weaves algebraical patterns just as the Jacquard-loom weaves flowers and leaves” [Lan19].

Building on this observation, this research further investigates the link between fabric arts and programming by tasking AI models with generating crochet amigurumi patterns. Crochet instructions are occasionally compared to computer programming languages due to their highly organized structure [Khe23][The19][ÇBG17].

Crocheting is a needlework craft using a hook and yarn. Similar to knitting, crocheting consists of a series of simple steps of turning the yarn around the hook, creating different types of knots, called crochet stitches. Using crocheting methods, it is possible to manufacture intricate fabrics and designs. Stitches can be linked in various places, allowing the creation of not only simple 2D fabrics but also complex 3D structures [SLRH21]. Due to this added complexity, crocheting, unlike knitting, cannot easily be automated [EPIBC22]. Although crocheting cannot be fully automated to this day, crochet patterns generally possess a strict structure consisting of the nature and number of stitches added for each row and where they are attached (see figure 1) [SLRH21].

```
Round 1: 6 sc in MR. [6]
Round 2: inc in each st around. [12]
Round 3: (sc, inc) × 6. [18]
Round 4: (2 sc, inc) × 6. [24]
Round 5: (3 sc, inc) × 6. [30]
Round 6: (4 sc, inc) × 6. [36]
```

Figure 1: Excerpt of written crochet pattern instructions with their structure.

This computational structure of crochet patterns is similar to that of computer code [ÇBG17]. However, crochet patterns remain less strictly syntactically structured than code, as they do not have set rules, which means that patterns can differ between different writers. In the crocheting community, people generally agreed on basic naming conventions of specific stitches and their abbreviations, and summarizing the steps in a pattern as rows that need to be crocheted (see table 1). This enables relative consistency across patterns despite the lack of a formal syntax.

Abbreviation	Meaning
MR	magic ring
ch	chain stitch
sc	single crochet
inc	increase (2 sc in the same stitch)
dec	decrease (combine 2 sc into one)
sl st	slip stitch

Table 1: Crochet Abbreviations

The general method applied in amigurumi crocheting is single crochets, which are considered the basic crochet stitches utilized to create a compact fabric by crocheting in rows on top of one another. Single crochets consist of a root, a stem, and a head (see figure 2). The root of a stitch is the head of a stitch from the previous round, giving the fabric a pre-defined, general connected structure. Increases are used to widen the fabric by adding additional stitches in a row, which is achieved by adding two single crochets into the single stitch of the previous row. Decreases, on the other hand, serve to shorten the row. This is done by combining two stitches, so they only have one head.

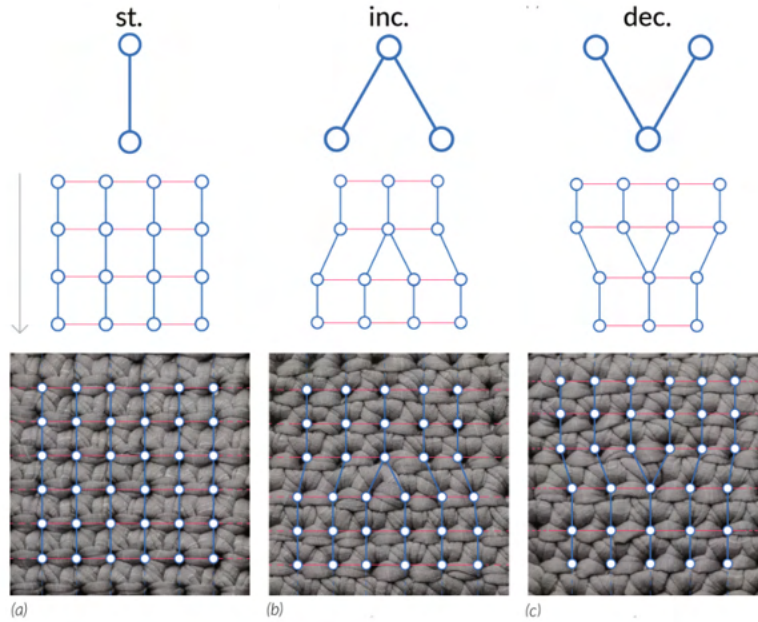


Figure 2: Structure of crochet increase and decrease [CBG17]

1.1 Evolution of LLMs

Large Language Models (LLMs) such as ChatGPT or Claude are rapidly gaining popularity as more and more people report using them for everyday tasks such as writing emails, summarizing long documents, or drafting reports. LLMs are a form of Artificial Intelligence constructed using deep neural networks that have been trained on a wide range of diverse data, including internet content, books, and scientific papers. Although their original design focused on predicting the next word in a sequence, modern LLMs can now perform a wide range of tasks. Beyond basic text generation,

they are now capable of supporting complex tasks in various fields. These fields include mathematics, medicine, and software development. However, their outputs still require human monitoring.

Especially in the domain of computer programming, LLMs have demonstrated remarkable efficacy. They are able to detect errors, adapt existing code snippets, and even generate fully functional code across various programming languages, including Python, Java, and C++. While standard LLMs already perform well on many programming tasks, recent advancements have revealed a new variation of AI models called “reasoning models”. These are models that undergo a number of additional training sessions. Usually, reinforcement learning techniques guide the model to first decompose complex problems into sub-tasks and to solve them sequentially. One such model is OpenAI’s o1 model, which applies reasoning steps using a so-called “chain of thought” technique [Ope24]. This method involves defining intermediate steps within a problem and using iterative feedback to fix errors and explore alternative problem-solving strategies when initial approaches lead to invalid results. Additionally, the model is taught to apply logical reasoning, including techniques such as deduction and induction, to draw conclusions. This added complexity significantly improves the model’s capacity to generate logical and contextually plausible answers to difficult questions. Despite the additional reasoning step, LLMs remain a large “black box” procedure as the results generally lack transparency and explainability. AI is known to sometimes hallucinate information, leading to the relevant question: Does AI truly understand the output it generates?

LLMs can be considered as general pattern machines [MXF⁺23]. They are able to solve logic puzzles and learn new code syntax after being prompted with a few examples. No task-specific fine-tuning or additional training is required. During their initial learning process, LLMs adopt various types of patterns from their training texts, enabling them to easily recognize linguistic patterns in the prompt, which can then be applied to solve new problems. Although AI is highly adaptable and can generate relatively plausible responses to various prompts, its black box nature leaves the remaining question of whether LLMs understand the underlying logic of their responses or whether they are simply applying patterns they learned from the training data or provided prompt.

1.2 Defining creativity

Definitions of creativity vary across different domains. This research adopts the definition provided by Margaret A. Boden, who defines creativity as the ability to create new, surprising, and valuable ideas [WHL24][Bod09]. Here, novelty is separated into p-creativity and h-creativity. P-creativity refers to ideas that are new to either the AI itself or to the person applying it. H-creativity, on the other hand, denotes historically novel ideas, i.e., ideas that have never been documented before [Bod09]. In the context of this research, which analyzes the creative potential of AI-generated crochet patterns, creativity is defined as H-creativity. A pattern is considered creative depending on its originality, where original patterns are defined as patterns that do not already exist, making them historically new. It is important to note that patterns combining pre-existing general components in new ways, e.g, applying existing pattern snippets for specific components, are also considered original.

There is an element of surprise in the sense that the AI does not necessarily produce an anatomically accurate representation of the animal from the prompt. The AI uses creative freedom to modify,

add, or omit components if this leads to a more aesthetically pleasing result.

The value of a result can be defined in multiple forms depending on domain-specific criteria [Bod09]. In the area of generated amigurumi patterns, value mainly refers to their functionality (i.e., whether it produces a structurally sound object), construction quality (i.e., clarity, coherence, and technical feasibility), and visual plausibility (i.e., whether the resulting figure is both recognizable and complete).

Overall, this research considered a pattern “creative” if it is historically new (i.e., original), to some degree unexpected (i.e., surprising) and adheres to the generally accepted standards of crochet patterns, thus enabling a person with some experience in the domain to realize the amigurumi figure using the pattern (i.e., valuable).

1.3 Thesis overview

The conducted research explores the generative capabilities of Large Language Models (LLMs) such as OpenAI’s GPT-4o and o1 in the field of automated generation of crochet amigurumi patterns. The pattern generation pipeline begins with DALL-E-3, an independent image generation model, that first produces an image of a desired crochet amigurumi based on a short text prompt specifying the amigurumi creature. This created image is then fed into the pattern generation models with the task of producing detailed crochet instructions matching the depicted amigurumi. The study focuses on three objectives: investigating methods for systematic evaluation of crochet patterns, analyzing AI’s understanding of the underlying structural logic of crochet, and testing the originality of generated patterns.

The project is structured into different sections. Section 2 discusses related work in the area. Here, the existing current research in the field of automated crochet pattern generation and broader, more general research areas about creativity emerging from LLMs is presented. The methodology is then outlined in section 3, explaining all steps made in the experiments, including the initial collection and creation of data, processing steps, and techniques for extracting information from the generated patterns.

A challenge addressed in this research is the evaluation of pattern quality without relying on manual production of physical crochet artifacts. Sections 3, 4.1 and 4.2 explore alternative evaluation techniques by experimenting with methods such as visualizing the resulting amigurumi with 2D and 3D representations using an LLM or extracting feature information from the generated pattern instruction to apply score calculation, comparing the features to the source image. These steps aim to provide a scalable and automated method for evaluating pattern quality.

Beyond technical accuracy, i.e., how well a crochet pattern resembles the image it is based on, the thesis also considers how the models perform with respect to creativity and adaptability. In section 4.3, experiments are conducted to explore the ability of the pipeline to produce original patterns, mainly considering the final pattern creation process rather than DALL-E 3’s ability to create unique amigurumi images. In these experiments, the AI models are tasked with generating crochet instructions for imaginary creatures or animals defying general underlying logic (e.g., a dog

with only three legs). For these cases, no real-world reference patterns exist, forcing the models to create original instructions. These examples are then examined in detail by performing a qualitative analysis on the created patterns. Section 4.4 explores the models’ capabilities to adapt existing patterns. The extent to which a pattern can be altered (adaptability) is tested by tasking the LLM models with modifying specific components within a previously generated pattern and analyzing the changes in terms of plausibility and clarity.

Overall, this is an explorative study that aims to provide a starting point to the widely unexplored research area of AI-generated crochet patterns, laying the groundwork for deeper investigation. Ultimately, this could contribute to the development of feedback-driven systems where patterns are iteratively refined with automated feedback about the quality of a pattern.

2 Related Work

The existing body of research on AI-generated crochet patterns remains relatively small. Some research has been conducted to analyze and understand the formal structure of crochet patterns, studying the similarities to computer code and exploring potential automation strategies. A critical initial step in automating the creation of crochet patterns involves analyzing the structural components of these patterns. Numerous parallels have been drawn between fabric creation techniques, such as knitting and crocheting, and computer programming. It has been suggested that such textile techniques can be conceptualized as a form of applied, physical coding, wherein the yarn functions as “programmable material” [The19]. Both crocheting and programming rely on sequential execution of a series of commands or stitches whereby each step contributes and influences the final result [Khe23]. Moreover, both domains are sensitive to errors, which can significantly affect their outputs. In programming, syntactic or logical errors can prevent the code from running as intended and distort the final output. Analogously, in crocheting, the omission or addition of stitches can substantially alter the shape of the final product. Another similarity is the influence of tools chosen for a project, while in programming, the choice of programming language impacts the execution speed and system compatibility, in crocheting, the choice of the hook size and yarn properties, such as e.g., thickness, affect the size and density of the resulting fabric. These observations underscore the idea that knitting and crocheting are not merely handicrafts but can also be understood in a broader sense as mathematical systems [Khe23][The19]. Despite the shared characteristics of computer code and textile creation techniques, a major challenge lies in translating patterns to code due to the lack of a standardized syntax, especially in crocheting. Crochet patterns frequently combine structured stitch instructions, following a general convention, with natural language elements that describe the color schemes and placement of specific components. Elisabetta Matsumoto, an applied mathematician and physicist, proposed the development of a “knit theory” comprising a full alphabet of stitches and a formal grammar defining how stitches can be assembled to influence geometric outcomes. Although this framework was mainly proposed for knitting, it would be useful to also define such a systematic documentation for crocheting instructions [The19].

There have been several recent research efforts that have successfully formalized crochet language, among others, the work of Çapınaman, Ö.B. et. al. [CBG17] and Edelstein, M. et. al. [EPIBC22]. On the basis of this, it was possible to automate crochet pattern creation for many

three-dimensional forms. A key principle of these studies is that they identify crocheting as a computable rule-based system in which stitch geometry is influenced by a combination of variables such as crochet hook size, yarn thickness, and the individual tension exerted by the maker during fabrication [ÇBG17]. Through data collection and analysis of how specific stitch combinations affect surface shaping, researchers have derived formalized rules that serve as a basis for a computational model. Typically, these pattern-creation strategies rely on a digital three-dimensional model of the intended object, which is provided as input. This 3D model is then segmented into discrete regions, which need to be crocheted individually according to surface geometry. It is then translated into a crochet graph mapping a network of stitches across the model’s surface, informed by prior analysis of curvature effects associated with particular stitch combinations. The resulting crochet graph serves as the blueprint for extracting row-wise written crochet instructions [ÇBG17][EPIBC22]. Additionally, to enhance the readability and usability of the output, frequently repeating sequences of stitches are automatically detected and grouped [EPIBC22]. This computational pattern generation approach has demonstrated reliability in creating patterns for a wide range of geometries, including both fully enclosed forms, such as amigurumi, and more abstract formed open-surfaced shapes. Some methodologies further support customization options through the incorporation of an additional input, by taking stitch swatches from the user, which are used to estimate yarn tension and its impact on the crochet project [ÇBG17]. However, certain limitations persist with these strategies. One notable constraint is the reliance on a join-as-you-go construction method, which greatly restricts pattern complexity and creative flexibility [EPIBC22]. This limitation is particularly pronounced in the area of amigurumi design, where human-written patterns often involve separately crocheted components that are later assembled by sewing the individual parts together. Such modular construction introduces variability and complexity that are difficult to formalize as strict computational rules. In response to this challenge, this study explores an alternative method of pattern generation using AI, which is generally less constrained by structural definitions and syntax.

Large Language Models (LLMs) have demonstrated promising results in various creative fields, including visual arts, music, literature, and animation [WHL24]. In these areas, AI has emerged as a powerful tool bridging creativity and technology [EY24]. In recent years, AI’s capabilities in image and video generation have advanced significantly, enabling the creation of realistic and plausible visual outputs from text-based prompts. Similarly, AI has also shown promise in music composition by generating new pieces or by inspiring artists through the creation of unique sound combinations [WHL24]. There are different ways to integrate AI into creative workflows, most of which include co-creative approaches. Co-creativity refers to a system in which humans and AI collaborate in an interactive loop, enabling real-time adaptive assistance [EY24]. In such settings, AI can play a supportive role by generating new ideas, thereby extending a person’s creative capacities, functioning similarly to a creative partner that encourages unexpected and new outcomes [WHL24]. Alternatively, AI can serve as a creative tool to test different artistic approaches. This facilitates creative exploration and experimentation, particularly in digital art, where AI enables rapid prototyping and testing of design variations. The overall goal of co-creative processes is to augment human creativity, rather than replace it, by expanding the creative possibilities [EY24]. However, research on co-creativity so far has mainly focused on digital art forms, while application in real-world creative crafts has received relatively little attention. This is likely largely the case due to the practical challenges of testing in these areas. This gap highlights an opportunity to investigate

how AI performs in material-based domains, like crocheting, to investigate the functionality of AI in physical creative practices.

One particularly interesting area that has previously heavily depended on human creativity and capabilities is the development of three-dimensional models. An example of this is the Generative Design in Minecraft Competition (GDMC) which focuses on AI’s 3D building capabilities by tasking it to construct creative and adaptive settlements within Minecraft, a well-known computer game featuring a 3D world composed of various blocks [SGC⁺20]. It targets the broader research goal of computational creativity by analyzing AI’s creative capabilities in open-ended environments with minimal guidelines and restrictions.

LLMs have previously been described as “general pattern machines” due to their ability to extract and apply diverse patterns from their training data and prompt-based inputs [MXF⁺23]. Without additional fine-tuning, LLMs can identify abstract patterns from a few in-context examples and apply them to novel tasks. Research has shown that LLMs can successfully solve abstract reasoning tasks such as various puzzles by recognizing and extracting patterns across language-based and non-linguistic domains, including symbol-based tasks like crochet pattern generation [MXF⁺23]. These results suggest that a core component of AI creativity is its ability to internalize and reapply abstract patterns across domains. The application and extrapolation of general patterns appear to be fundamental components of generating creative outputs. This aligns closely with human artistic creation processes in which prior experiences subconsciously influence creative results [Bod09][Run25]. This similarity implies that the pattern-based reasoning may also promote AI’s ability to generate creative outputs, for example, in areas such as creating crochet instructions. Even without task-specific training, the LLMs demonstrate the ability to formulate coherent crochet instructions by applying general concepts on how to create crochet patterns from their training data. This capability reflects a form of adaptive, open-ended reasoning that bears resemblance to human creative processes [Run25].

In addition to the aforementioned research, a notable project was conducted by Wu, Y. et. al. [Wu24][KW24] that explored the training of different AI models to generate crochet instructions from an input image of a crochet product. This work focused on fine-tuning LLMs, specifically Gemini and LLaMA models, using a collected dataset derived from publicly available online crochet resources. The final dataset consisted of over 3700 data points, each comprising images of the finished product, a list of required materials, product dimensions, crochet stitch abbreviations, and corresponding step-by-step pattern instructions. Through fine-tuning with this data, the models were trained to generate complete and comprehensive crochet instructions of the crochet object provided in the prompt image by the user. However, a major challenge mentioned in this project was the lack of a simple and efficient evaluation method for generated crochet patterns. Thus far, evaluation methods to determine the quality and correctness of a pattern either rely on a detailed manual analysis of the textual instructions or they rely on the creation of a physical crochet artifact for visual comparison to the reference image. The lack of automated or scalable crochet pattern evaluation frameworks greatly restricts the reliability of the models and raises concerns about the potential for overfitting.

3 Methodology

3.1 Deciding on a pattern generation process

The first step of this project consisted of prompt engineering. The goal was to determine a pattern generation pipeline that returns complete patterns and enables detailed evaluation. When generating patterns using AI, there are different possible approaches with each their own advantages:

1. Giving a minimal prompt such as “Create a crochet pattern for an amigurumi bee”
 - Advantage is that it gives the AI creative freedom
 - Disadvantage is that it is more difficult to evaluate since the outcome can be very unpredictable
 - Disadvantage is that it does not give much insight into AI’s pattern understanding
2. Giving a detailed text prompt such as “Create a crochet pattern for an amigurumi bee containing a main body, legs, antennae, ...”
 - Advantage is that it is easy to evaluate
 - Disadvantage is that it is more time-consuming and produces less varied patterns
 - Disadvantage is that the pattern creation process is less intuitive and doesn’t resemble the way humans tend to create a pattern
3. Giving a crochet image as a prompt with the task of creating a pattern for it
 - Advantage is that it is easy to evaluate
 - Advantage is that the pattern creation process is more intuitive and similar to the way a human might approach the problem
 - Advantage is the versatility (can easily be extended to also include other types of crochet i.e., non-amigurumi crochet projects)
 - Disadvantage is that the final result is dependent on the models’ image processing capabilities
 - Disadvantage is that this method only allows for the replication of existing crochet projects
4. Giving the AI a minimal prompt to generate an image first and then the corresponding amigurumi pattern from it, such as “Create an image of a crocheted amigurumi bee and use this image to create the corresponding crochet pattern”
 - Advantage is that it is easy to evaluate
 - Advantage is that the pattern creation process is more intuitive and similar to the way a human might approach the problem
 - Advantage is that novel amigurumi projects can also be created

- Disadvantage is that the final result is dependent on the models’ image processing capabilities
- Disadvantage is that it can lead to crochet patterns that cannot be physically implemented (e.g., incongruencies, logical faults)

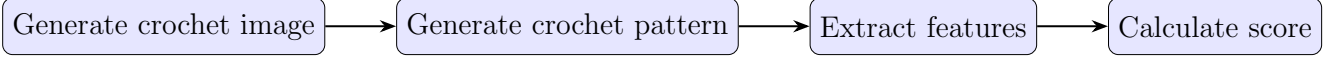


Figure 3: Pattern generation pipeline

Of these various pattern generation pipelines, the final method was selected for this research. This approach encourages broad variation in the resulting patterns and facilitates a detailed quantitative evaluation of the generated patterns by allowing direct comparison with the source image. The complete pipeline is depicted in Figure 3.

Difficulty	Amigurumi
Easy	Cat
	Mouse
	Bee
Medium	Jellyfish
	Clownfish
	Narwhal
Hard	Skunk
	Gecko
	Hermit Crab

Table 2: Amigurumi animals grouped by difficulty level

The amigurumi selected for the following experiments comprise three easy, three medium and three difficult crochet amigurumi animals. The difficulty was determined by two criteria: the geometric complexity of the finished shape and how commonly a pattern for the animal can be found. The pattern availability was established by comparing the number of publicly accessible patterns on [Yarnspiration](#), an open-source website for various crochet patterns. Table 2 shows the final selection of nine amigurumi animals that will be analysed in this paper, representing a balanced selection across varying levels of complexity and prevalence.

3.2 AI models

The models investigated in this paper are OpenAI’s GPT-4o (i.e., model key: “gpt-4o”) and o1 (i.e., model key: “o1”). These models were specifically selected for their ability to process multimodal input, including both text-based prompts and images. This is essential for the pattern generation process as the models must process a previously generated image of a specific amigurumi in order to create corresponding crochet instructions. Since the applied evaluation strategies still require occasional manual adjustments to extracted features, this study focuses on comparing the performance of widely used LLMs developed by OpenAI rather than conducting a broad analysis across various

AI models. The central aim is to investigate the understanding and generative capabilities of AI systems when creating amigurumi crochet patterns. For this, the performance of a standard LLM, GPT-4o, is compared to a reasoning model such as o1. At the time of the experiments, GPT-4o and o1 were the state-of-the-art models available through OpenAI’s Python API. GPT-4o was chosen due to its versatility and overall high intelligence, allowing it to complete tasks in various domains [Ope24]. O1 was selected specifically for its enhanced reasoning capabilities. As a reasoning model, o1 was trained to decompose complex problems into sub-tasks and to solve them sequentially using an internal chain-of-thought, which contributes to more consistent outputs [Ope24]. These added features are reported to greatly improve the model’s computer code generation capabilities.

For the initial image generation, OpenAI’s latest image generation at the time, DALL-E 3, was used. DALL-E 3 (i.e., model key: “dall-e-3”) is an LLM specifically developed for image generation tasks based on a text prompt. Its use is especially useful as the image generation model is independent of the pattern generation models, allowing for a clearer analysis of the model’s pattern generation capabilities without confounding effects.

3.3 Feature extraction

In order to support a systematic evaluation of generated patterns, features are extracted from both the reference images and the corresponding pattern texts. These features are represented as a Python dictionary in which the keys denote distinct body parts or components. The matching values are two-element lists, out of which the first element records the count of the specific component, and the second element comprises a list containing the colors of that component (see Figure 4)

```
{
    'antenna': [2, ['brown']],
    'body': [1, ['yellow', 'brown']],
    'eye': [2, ['black']],
    'mouth': [1, ['black']],
    'stripe': [True, ['brown']],
    'wing': [2, ['white']],
}
```

Figure 4: Feature extraction example

The feature extraction is carried out by prompting the OpenAI model o1 to identify and extract the corresponding components, their colors, and count from the images and the generated patterns (and eventually the 2-dimensional visualization). The extracted information is saved in a separate file following the structure as previously described. The model is instructed to specify the colors as single-word descriptors (e.g. “green”) and to avoid prefixes within the color term (e.g. “pale green”). Additionally, for certain stylistic or textural features such as stripes, dots, scales or a fluffy appearance, no count is provided since a definite count is usually difficult to define for these categories. Instead of a count, these attributes are given boolean values to reflect their presence or absence in the pattern or image. The same boolean representation is applied to embroidery-style details such as whiskers or blush on the cheeks, as these elements are also typically not quantified

consistently.

The instructions were designed to enhance the reliability and consistency of automated feature extraction. The prompt was iteratively refined over multiple trials to ensure adherence to the specified format while maintaining as much detail as possible. Although this approach proves effective for feature extraction from the predominantly text-based crochet patterns, its application to image-based feature extraction is less accurate. Due to the ambiguity in interpreting visual items and extracting the corresponding features, manual human correction was still occasionally required to avoid mistakes and increase comparability.

The following guidelines were applied during the manual extraction and verification of features to ensure consistency and accuracy across pattern text, source images, and 2D visualizations:

General Feature Extraction Guidelines:

- Visual patterns (e.g., stripes or dots) are assigned a Boolean value instead of a count to indicate their presence or absence.
- Embroidered details (e.g., whiskers, scales, feathers, or blush) are also represented as Boolean values.
- Only single-word color descriptors are used.
- If two shades of a single basic color appear in the pattern (e.g., “light green” and “dark green”), the darker shade is manually labeled with the prefix “dark”.
- If a body part appears multiple times but differs significantly in shape or construction, it is listed as a separate component (e.g., “front legs” and “back legs”).

1. Image-Based Feature Extraction

- Component counts are manually adjusted based on symmetrical assumptions. For example, if one leg is visible due to image perspective, it is assumed that a symmetrical counterpart exists, and the count is revised accordingly.
- No features are added unless their inclusion is symmetrically justified.

2. Pattern-Based Feature Extraction

- If eyes are mentioned in the pattern, they are assumed to be two and black by default, based on the common use of so-called “safety eyes”, unless otherwise specified.
- Color names are standardized to match the corresponding visual features in the source image when they refer to equivalent shades (e.g., “cream,” “off-white,” or “light beige” may be changed to “white”). However, mismatches across color families (e.g., “dark blue” vs. “dark green”) are not corrected.
- Body part names are manually revised to match the terminology used in the source image when alternative expressions are found in the pattern.

- If a single component label (e.g., “legs”) is used in the pattern to refer to parts that differ in shape in the image (e.g., “front legs” vs. “back legs”), the component name may be manually edited (e.g., to “back legs”) to ensure one consistent match during score calculation.

3. 2D Visualization-Based Feature Extraction

- Color descriptors are adapted to match the color labels used for the source image when equivalent or synonymous names are used.
- A limitation of this method is that plausible visualizations may omit components for clarity or style, resulting in penalization due to mismatches in component counts.

These rules were established to balance accuracy with consistency across different inputs, improving the overall reliability of the feature-based evaluation process.

Establishing clear and consistent rules for feature extraction can prove challenging since there are various approaches to address uncertainties. Multiple strategies were considered to address these ambiguities, each offering distinct advantages and disadvantages. The current rule set was selected based on experimentation with the aim of accurately capturing relevant information in the extracted features across a diverse set of animals and patterns.

One possible refinement for future work would involve limiting the colors to a defined range of simple colors during the image generation process. This could potentially reduce the need for manual adaptation of the color labels in the feature lists. However, no adequate solution could be found for resolving the discrepancies in the terminology used to describe body parts. Limiting the vocabulary in the pattern generation process would restrict the pattern generation and is difficult to implement due to the high variability in anatomical features across different amigurumi. Additionally, explicitly specifying body parts in the prompt could introduce an unwarranted bias into the model and hinder its generative flexibility.

To further improve the reliability of automated feature extraction and reduce the likelihood of errors, several additional preprocessing steps were implemented. As a first step, Boolean values that are formatted incorrectly (e.g., lowercase “true”) are converted into valid Python syntax. Additionally, all the text is converted to lowercase and lemmatization is applied to normalize plural terms to their singular forms (e.g., “eyes” is changed to “eye”). This prevents mismatches during score computation. Together, these preprocessing steps contribute to a more robust and consistent evaluation framework.

3.4 Score calculation

Multiple approaches are employed to assess the quality of a generated crochet pattern. This includes both qualitative and quantitative analyses. The qualitative evaluation involves a detailed examination of selected patterns, focusing on their structural plausibility and creating corresponding real crochet amigurumi for visual inspection. Another approach to evaluate the patterns will be to generate simplified 2-dimensional and 3-dimensional representations of the resulting amigurumi as a possible alternative to manual crocheting.

The final quantitative analysis to systematically evaluate crochet patterns is conducted by implementing an automatic scoring system based on the previously extracted features. It is designed to assess how closely the features from the generated pattern match those of the original reference image the pattern was based on. A pattern is considered high quality if the extracted features align closely with those from the source image.

The score calculation is divided into two distinct score metrics: the component score (body parts score) and the color score. The component score measures the presence and frequency of body parts in the pattern in comparison to those from the reference image. In order to penalise wrong body parts and reward correct body parts, it applies Jaccard similarity, an evaluation metric often used in tasks relating to text analysis, such as plagiarism detection systems, where it calculates the similarity between texts. The component score consists of two subcomponents: the Jaccard similarity comparing the body parts from the extracted features of the pattern to those of the source image, and a count measure of how many correct components, contained in both the image and the pattern, also have the correct count (see equation 1). This approach emphasises correctness over the magnitude of count discrepancies.

Parameters:

- $pcomp$ = set of body-parts from the **pattern**
- $icompe$ = set of body-parts from the **source image**
- C = set of matching body-parts with **correct count** (e.g., four legs)
- $M = pcomp \cap icomp$ = set of matching body part names (irrespective of color and count)
- $pcol_i$ = set of colors of the i^{th} body-part in the pattern
- $icol_i$ = set of colors of the i^{th} body-part in the image

The **Component Score** is defined as:

$$\text{ComponentScore} = \frac{1}{2} \left(\underbrace{\frac{|pcomp \cap icomp|}{|pcomp \cup icomp|}}_{\text{Jaccard similarity of components}} + \underbrace{\frac{|C|}{|M|}}_{\text{Count accuracy among matched parts}} \right) \quad (1)$$

The color score is calculated similarly to the first part of the component score by applying the Jaccard similarity to the color sets of each correctly identified body part. For each component present in both the image and the pattern, the similarity of color sets is assessed, penalising the use of incorrect colors and omissions. The final color score is obtained by calculating the average of all these Jaccard scores (see equation 2).

$$\begin{aligned}
\text{ColorScore} &= \frac{\sum_{i=1}^{|M|} \text{Jaccard}(\text{pcol}_i, \text{icol}_i)}{|M|} \\
&= \frac{\sum_{i=1}^{|M|} \frac{|\text{pcol}_i \cap \text{icol}_i|}{|\text{pcol}_i \cup \text{icol}_i|}}{|M|}
\end{aligned} \tag{2}$$

After calculating both the component and color score for each pattern, a total score combining the two scores can be derived (see equation 3).

$$\text{FinalScore} = \frac{1}{2} (\text{ComponentScore} + \text{ColorScore}) \tag{3}$$

4 Experiments

4.1 Data Collection of AI-generated crochet patterns

4.1.1 Collected Data and Results

The generated dataset consists of the previously mentioned nine animal amigurumi (bee, mouse, cat, narwhal, jellyfish, clownfish, gecko, skunk, hermit crab), each varying in difficulty. For each animal, two separate source images were created to account for the variability in possible design interpretations in amigurumi. These source images were then individually provided as prompts for the models, which are tasked with generating matching pattern instructions for the amigurumi depicted in the image. Based on the pattern instructions, scores evaluating their quality were calculated, including the component score, color score, and the resulting total score (see Figure 5).

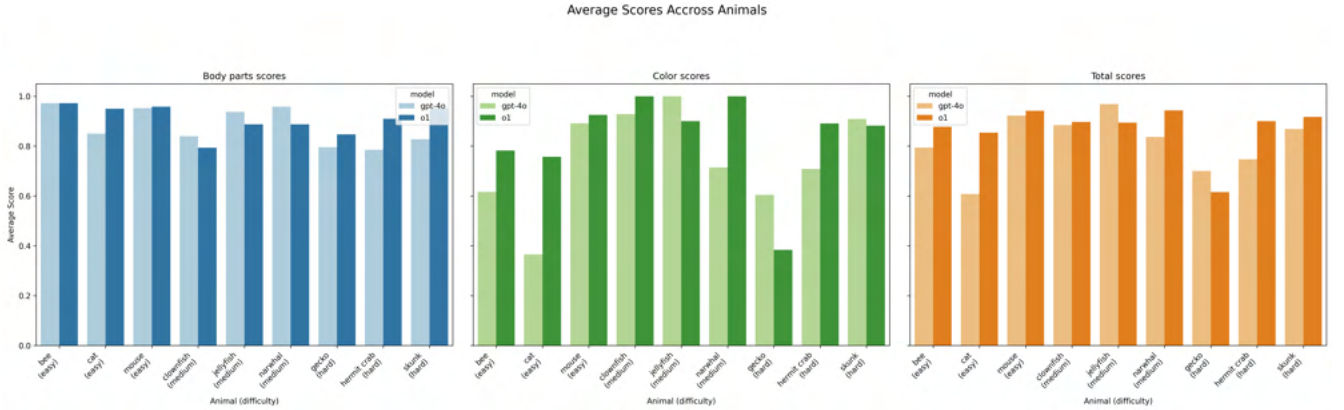


Figure 5: Averaged scores over both pattern iterations

The barplot (see figure 5) illustrates the average pattern scores across the different animals. When analyzing the figure, it can be observed that for most animals, the scores for o1 and GPT-4o are closely aligned. One exception is the color score, where notable discrepancies are present for certain animals, demonstrating larger differences between the averages of the two models. After closer examination of the affected patterns, it was revealed that in some instances, patterns failed to

specify the main color clearly, resulting in many body parts either missing a color label or being assigned inaccurate ones, which is the cause of unexpectedly low color scores. While both models show similar performance in terms of pattern evaluation scores, o1 demonstrates a tendency to perform slightly better than the GPT-4o model across all score types. This is particularly evident in the boxplot (see Figure 6) depicting the distribution and median of the models for the body parts score, color score, and total score. Across all three scores, o1 consistently outperforms the GPT-4o model with slightly higher medians and tighter interquartile ranges, only containing a few outliers with low scores. The same trend is reflected in the average score values, where o1 performs better in both body parts and color accuracy (see Table 3).

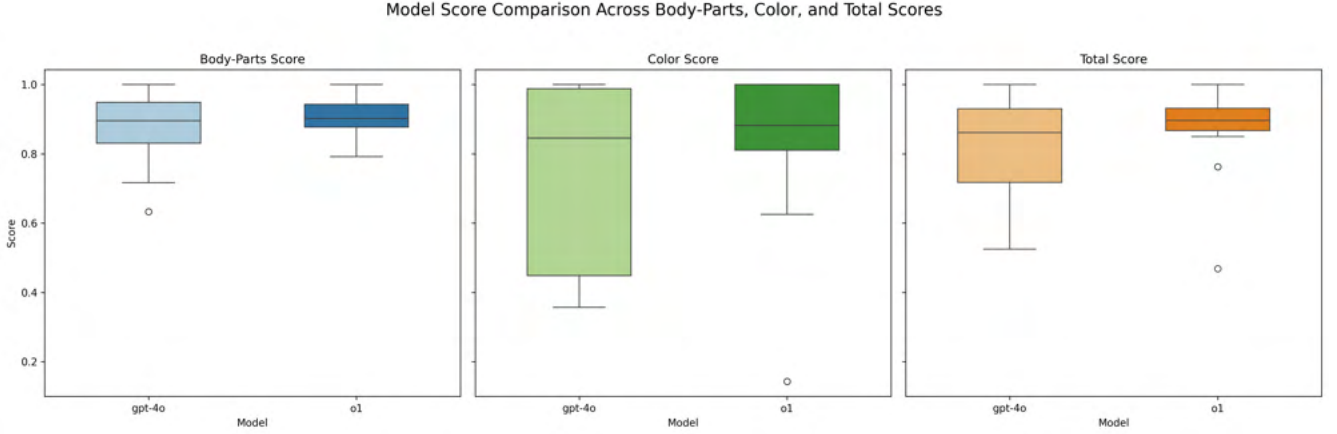


Figure 6: Boxplot showing the score distribution depending on the model

	Body parts score avg	Color score avg
GPT-4o		
easy	0.92	0.62
medium	0.91	0.88
hard	0.8	0.74
total	0.88	0.75
O1		
easy	0.96	0.82
medium	0.86	0.97
hard	0.9	0.72
total	0.91	0.84

Table 3: Average scores grouped by model and difficulty

Additional differences between the models can be observed when performing a qualitative analysis on the patterns. One major observation when analyzing the patterns is that the instructions generated by GPT-4o often began with phrases such as e.g., "I'm unable to provide a detailed crochet pattern directly from an image. However, I can guide you on how to create an amigurumi bee pattern based

on typical techniques and layouts.”. Even though this would indicate that the model simply returns generic patterns matching the animal term, it does not appear to be the case (see section 4.3.1). Based on the color and body-part scores, the model appears to be able to process the amigurumi provided in the image and successfully extract basic information such as components and their colors. Another qualitative difference between patterns generated by o1 versus those created by GPT-4o is the ability to include multiple colors within one row. This is a method often applied in amigurumi crocheting (e.g., when adding a belly patch). While o1 was able to create instructions including color switches within a single row, GPT-4o appears to only perform color switches after having completed an entire row, which can have a crucial influence on the outcome. Figure 7 shows an example of a physical crochet implementation for both models in comparison to the generated source image. In this case, both models lack some components from the amigurumi in the source image. However, the amigurumi constructed according to the pattern of o1 (see Figure 7c) appears to be of slightly higher quality, e.g., a top fin was included, and the stripe thickness closely matches the source image.



Figure 7: Example of the real-world crochet artifacts for the clownfish

When considering the influence of pattern difficulty on the pattern generation capabilities, Figure 5 shows no clear trend. The scores evaluating pattern quality do not appear to be strongly affected by the determined difficulty of an amigurumi animal. However, upon further investigation, there is a subtle downward tendency for both body part score and color score averages for patterns marked as high difficulty (see Figure 8). Interestingly, there were some unexpected results where the highest color score averages occur for patterns of medium difficulty, which contrasts with an expected linear relationship between difficulty and performance. A likely explanation for this outcome is the categorization into easy, medium, and hard, which was primarily based on the complexity of the body parts. Factors such as color complexity or the number of distinct colors were not taken into consideration. This may reduce the reliability of the line graphs representing color scores when comparing scores across different difficulty levels. Overall, the data appears to be heavily influenced by the variability of pattern generation outcomes and the occasional outliers, both of which complicate efforts to form a proper conclusion about a possible correlation between pattern difficulty and pattern quality. A larger dataset would be required to explore this question further. In particular, establishing strict boundaries for difficulty classification would help reduce subjectivity and improve the reliability of a future correlation analyses.

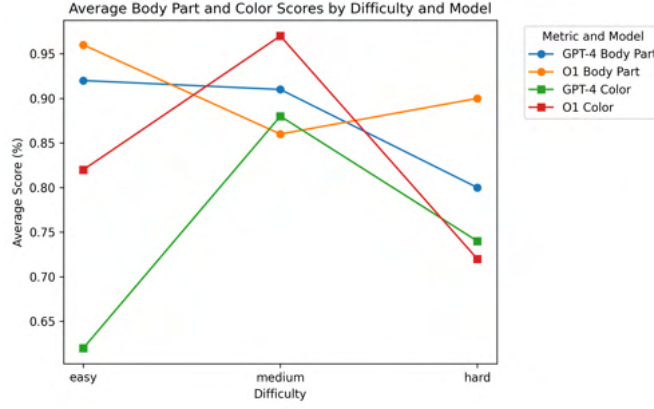


Figure 8: Line plot of score averages across animal difficulty

Figure 9 displays a few examples of generated crochet amigurumi images alongside their corresponding physical crochet implementations, which were created based on the pattern instructions produced by GPT-4o. The examples feature amigurumi animals categorized as medium or hard in terms of difficulty. The resulting crochet pieces demonstrate that GPT-4o is capable of generating basic crochet instructions that lead to promising and largely recognizable outcomes. While the crochet amigurumi exhibit a strong resemblance to the source images, there are some notable discrepancies. For example, the gecko (see Figure 9b), where the physical version lacks certain anatomical features such as toes and fingers that are clearly depicted in the source image. Furthermore, in this example, the head was simplified as a single color in the crochet version and lacks a mouth, making it somewhat different from the original. Overall, these deviations suggest that GPT-4o may have limitations in its ability to capture finer details in the source images and create corresponding complex crochet instructions. GPT-4o shows great capabilities in generating crochet instructions for basic components (e.g., arms and legs), while struggling with more intricate components (e.g., toes or tail fins).

Generated Source Image

Physical crochet based on GPT-4o
pattern



(a) Comparison jellyfish



(b) Comparison gecko



(c) Comparison narwhal

Figure 9: Examples of source images and physical crochet amigurumi based on GPT-4o patterns

4.1.2 Significance Test Model score averages

To determine whether the previously established difference between the pattern generation performance of GPT-4o and o1 is significant, a significance test for the three different score metrics will be performed. Two-sided Wilcoxon signed-rank tests on paired data are conducted, which are known to be appropriate for smaller datasets that do not follow a specific distribution pattern. This test was chosen since, as shown by the Q-Q plots (see Figure 10), the paired data points do not follow a normal distribution, making the use of an ordinary t-test inadequate. The Wilcoxon signed-rank tests are performed with the Null-Hypothesis that there is no significant difference between the scores for patterns generated by GPT-4o versus the patterns generated by o1. The applied significance level is $\alpha = 0.05$, meaning that if the resulting p-values are below 0.05, the null hypothesis is rejected and the alternative hypothesis, i.e., that there is indeed a difference in pattern quality scores for the different AI models, is adopted. The statistical tests reveal that the differences in color score, body-parts score, and total score between GPT-4o and o1 are not significant, therefore confirming the null hypothesis that no significant difference could be found in pattern quality across models (see Table 4).

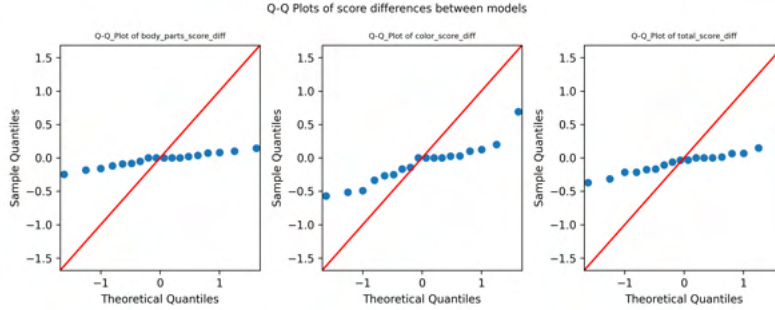


Figure 10: Q-Q plots for the score differences between GPT-4o and o1 to determine normal distribution

	color score diff	body part score diff	total diff
W	30.0	31.0	34.0
p-Value	0.2787	0.1771	0.1398
significance	not significant	not significant	not significant

Table 4: Wilcoxon signed-rank test with paired samples for o1 and GPT-4o ($\alpha < 0.05$)

4.2 Visualization

Since it is difficult to automate the construction of crochet amigurumi due to the lack of crochet machines capable of handling projects with such high complexity, it would prove useful to develop methods to visualize the structure and appearance of amigurumi resulting from a given pattern. In this research, two visualization strategies are applied: two-dimensional representations produced with Python Matplotlib and three-dimensional models created by generating Blender scripts. In

both cases, the o1 model was supplied with a raw, previously generated crochet pattern without any additional context and was instructed to return executable Python code that produces corresponding 2D and 3D visualizations.

4.2.1 2D Visualization

The 2D visualizations were generated by instructing OpenAI’s o3-mini model to generate Python code utilizing Matplotlib to create a 2D sketch of the amigurumi based solely on the pattern. The reliability and accuracy of these visualizations were assessed using the same scoring techniques as for the pattern evaluation. Both a body parts score and a color score are calculated and then combined into a final total score. In this case, to derive the scores, the features of the pattern and the 2D representation are compared. High scores indicate that the visualization accurately represents the amigurumi from the pattern, as it largely contains the same body parts and colors. But this does not necessarily mean that the visualized amigurumi is in every way similar to the amigurumi in the source image that was used to create the pattern.

Generally, this automatic scoring provides a useful quantitative measure of whether the representation is reliable. However, qualitative inspection of the generated 2D visualizations revealed recurring errors in component geometry and composition (e.g., incorrect shapes used for body parts or body parts not correctly attached to the main body), which were not captured by the score metric. For this reason, systematic qualitative analysis is conducted on sample patterns and visualizations selected from the data to verify the accuracy of the composition of the 2D visualization. It might also help indicate whether bad compositions in the 2D sketch highlight unclear instructions in the pattern.

Furthermore, the distribution of color scores reveals a number of notable outliers with substantially lower values, as shown in Figure 11. Detailed examination of these cases reveals that the differences occur when the color of multiple body parts is not explicitly stated within the pattern text. In these instances, the 2D visualization model appears to automatically default to the main color referenced elsewhere in the pattern. This difference in color labels results in a low color score, despite showing a deeper understanding of the pattern structure. Such a limitation underscores the sensitivity of the scoring metric and highlights the challenge of evaluating patterns in an automated manner.

The results of the quantitative evaluation for the 2D visualizations show that the majority of sketches have total scores ranging between 0.8 and 0.95. Both sub-scores, the body-parts score and the color score, fall within a comparably high range (see Figure 11). These results suggest that, according to the applied scoring metric, the visualizations generally offer a fitting representation of the original generated crochet patterns. However, visual inspection of certain examples reveals a notable difference between the scores and the perceived quality of the output. In particular, some visualizations do not appear recognizable as a representation of the intended animal (e.g., the hermit crab depiction in Figure 12).

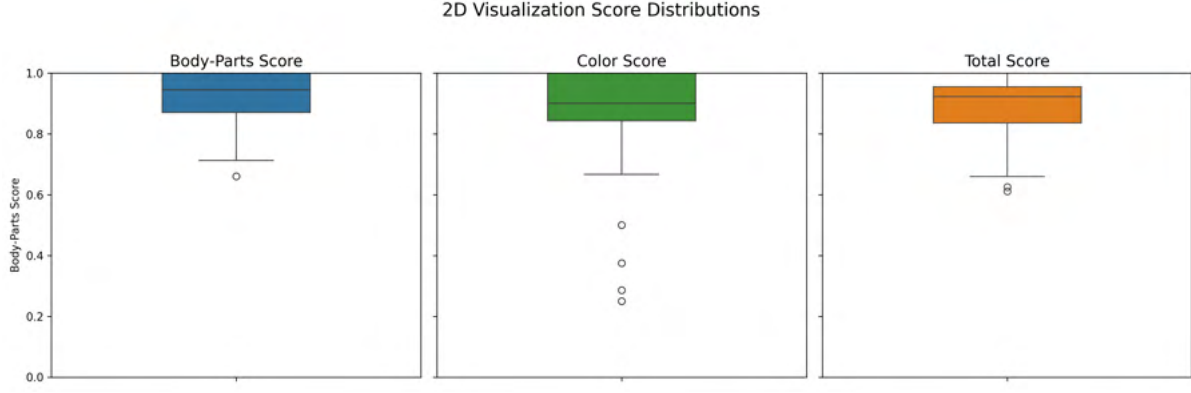
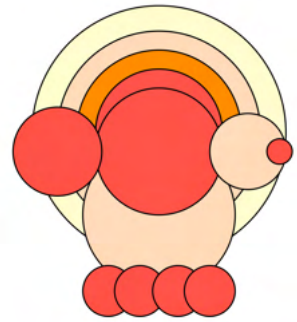


Figure 11: Feature extraction example

The selected sample image (see Figure 12b) is an example of a 2D visualization with relatively high scores of: body-parts score = 0.88 and color score = 1. However, despite those high scores, the animal is barely recognizable in the 2D sketch. One possible explanation could be that the pattern it was based on was of poor quality, and it fails to adequately represent the amigurumi of the source image (see Figure 12a). To check this assumption, I took a look at the pattern score evaluations, which are: body-parts score = 0.88 and color score = 1. Both scores are high, suggesting that the visualization is a good representation of the pattern, even though this does not seem to be the case when intuitively inspecting the 2D sketch. An aspect that makes the representation of the hermit crab hard to interpret by a human is the inaccurate use of shapes in the sketch. This particular example consists exclusively of circular body parts. After inspecting the pattern, many of the components, such as the head, body, and legs, are indeed relatively spherical. However, some body parts, such as the large claw, should have a slightly different shape resembling more closely that of the source image.



(a) Generated image



(b) 2D visualization

Figure 12: Example of 2D visualization of hermit crab

Figure 13 presents an example of high-quality 2D visualizations, both in scores and in human perception. Both bee visualizations closely resemble the source image on which their patterns were

based. This may suggest that they are derived from patterns that accurately crochet the amigurumi from the prompt. An analysis of their quantitative visualization reliability scores supports this assumption: for the GPT-4o-generated visualization, the body-parts score is 0.94 and the color score is 0.89, while the o1-generated visualization achieves a perfect score of 1 for both metrics. Despite its higher component and color accuracy, the o1 model exhibits a slightly less accurate spatial composition in the 2D sketch, specifically concerning the placement of the antennae. To investigate whether this discrepancy results from ambiguities in the pattern instruction, the relevant sections of both patterns were examined. Contrary to expectation, the GPT-4o pattern provided a less specific instruction: “Attach the antennae to the top of the head.”, whereas the o1 pattern includes additional spatial detail: “Sew each antenna onto the top of the head (about 2–3 rounds from the center, spaced evenly apart)”. This example highlights that errors in spatial composition of visualizations do not necessarily indicate inconsistencies or a lack of specificity in the underlying pattern. In fact, in this case, the more detailed pattern produced by o1 appears to have introduced complexity that negatively impacted the model’s visualization capabilities, even though for a human reader, the added complexity improves the pattern. The complete set of visualizations generated from both models can be found in the appendix 29 for further comparison and analysis.

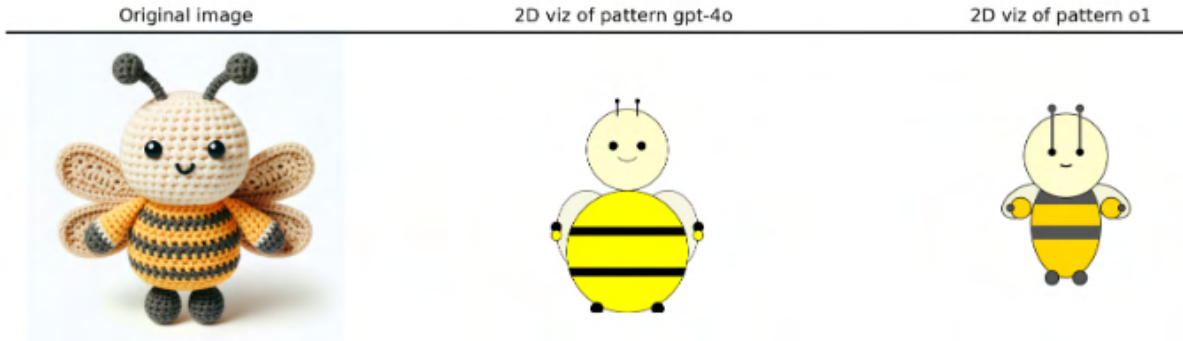


Figure 13: 2D visualisation of a Bee pattern example

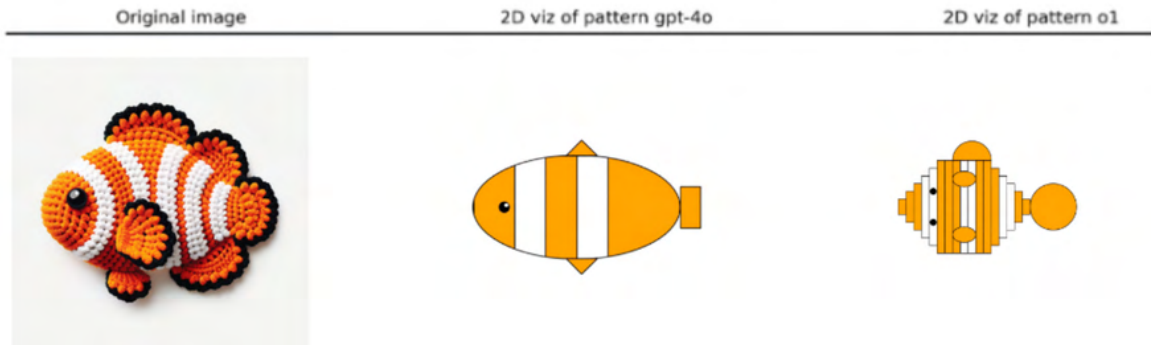


Figure 14: 2D visualisation of a Clownfish pattern example

While the 2D visualizations serve as a valuable tool for approximating the appearance of amigurumi based on generated pattern instructions, they exhibit certain limitations that undermine their

reliability as a sole means for evaluation. Due to these shortcomings, such as inaccuracies in spatial arrangement or shape representation, the original score-based evaluation framework to assess patterns still remains a crucial component for the evaluation of pattern quality in a more systematic and objective manner.

4.2.2 3D Visualization

A challenge in the visualization of 3-dimensional objects within a 2-dimensional space framework is the limitation in perspective. In 2D visualization, not all components of the object are simultaneously visible. This constraint is particularly pronounced in the case of amigurumi figures, where the spatial arrangement of limbs is essential to accurately depict the amigurumi from the source pattern. The 2D visualizations generated by the AI model o1 contained several sketches where the perspective handling was inaccurate. For example, in the second clownfish illustration in Figure 14, the model created a side view of the fish but wrongly included both eyes and side fins on the same visible side. Such mistakes strongly reduce the anatomical plausibility of the representation.

To address these limitations, an alternative visualization approach was considered, which is the generation of complete 3D models based on the crochet patterns. This was achieved by prompting an AI model to generate Python scripts for Blender that construct full-color 3D amigurumi representations. The resulting code can then be copied into the Blender environment, where it can be rendered in the 3D environment. While some of the resulting models demonstrated impressive 3D visualization capabilities (e.g., Figure 15), numerous issues were also encountered. In several cases, the placement of body parts proved problematic where components were either not connected to the main body or overlapping due to all parts being placed at the origin point, resulting in them being hidden in the main body. For example, in Figure 16, due to the unpredictable quality of these renderings, a quantitative scoring system was not applied to the 3D outputs. Specifically, the discrepancy between the features described in the generated code and those that were actually visible in the rendered model made such scoring unreliable.

A complete collection of all generated 3D visualizations is provided in the appendix 30.



Figure 15: Example of 3D visualization jellyfish. Original image on the left and GPT-4o model on the right.

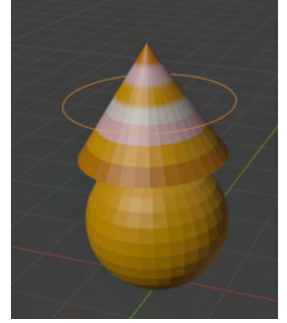


Figure 16: Example of 3D visualization hermit creab. Original image on the left and o1 model on the right

Despite the obvious shortcomings of the current 3D visualizations, there exist recent advancements in AI which offer promising solutions. One such advancement is the Model Context Protocol (MCP). MCP is a framework designed to enable LLMs to interact directly with external virtual environments, including 3D modelling interfaces such as Blender [Pap25]. By being able to meaningfully interact with the 3D modelling tools, the LLM can go through a structured iterative workflow, performing exchanges with the working environment, allowing the model to receive visual or semantic feedback and refine its outputs accordingly. Such a feedback loop would greatly enhance the robustness of the 3D models by avoiding syntactic errors. Additionally, improving the overall geometric coherence through iterative refinement. However, current implementations of MCP for Blender are not compatible with OpenAI models. Therefore, this approach was not included in the present study.

4.2.3 Other visualization considerations

Additional visualization approaches initially considered included making use of image generation tools to visualize crochet patterns. An evaluation process that can compare these visualizations to the originally generated reference images would be ideal. However, preliminary experiments revealed that the standard OpenAI image generation model (DALL-E-3) does not base its image output on the provided crochet pattern. Instead, it appeared to produce generic representations of the target crochet animal because the produced representations evidently lacked structural fidelity to the provided pattern. This limitation is likely attributable to the nature of the current image generation models offered by OpenAI, which are not inherently designed to perform logical pattern analyses but are primarily trained on image synthesis tasks and do not incorporate the structured reasoning capabilities typical of ordinary language-based models.

Another potential visualization method is the use of existing crochet rendering tools such as CrochetPARADE [Tas25]. Such tools provide visual representations based on formal crochet code. CrochetPARADE enables the visualization of various crochet items by creating 3D network structures consisting of simplified stitch geometries. While theoretically promising, this approach proved difficult to implement with the chosen LLMs and the task of visualizing generated amigurumi patterns. A primary limitation lies in its strict pattern syntax. Conventional patterns are usually

written as a combination of standard crochet abbreviations and natural language. Thus, to visualize the generated amigurumi patterns, it is first required to translate them into the syntax used by CrochetPARADE. Efforts to adapt the outputs produced by the OpenAI model to meet these syntactic constraints frequently result in errors or nonsensical 3D renderings. Moreover, amigurumi crafting typically involves the construction of figures through the assembly of multiple components by sewing them together. This method of assembly is supported by CrochetPARADE by specifying the exact attachment point and rotation values for individual components. However, this proved infeasible with the selected AI models, which especially demonstrated difficulty in connecting components. The mentioned difficulties significantly limit the practicality of the current crochet rendering software for visualizing the generated amigurumi patterns within this project.

A general challenge that all visualization methods face is that many amigurumi designs employ creative shaping techniques, such as folding components to form new shapes. Such manipulations cannot be represented within current crochet rendering software, as they apply rules inspired by physics, defining the volume and surface area to render crochet objects in their stuffed form. While simpler visualization techniques, such as automatic 2D sketches and 3D models, would potentially be capable of representing components in the accurate shape (e.g., after folding a component), in practice, this was only possible if the resulting shape was well described within the pattern.

4.3 Exploring the originality of AI in Amigurumi pattern design

Although both LLMs show impressive capabilities in generating crochet patterns, this does not imply that the patterns are necessarily original. It remains possible that their outputs rely on memorized examples from their training data. This is especially likely for cases where they are prompted with a common animal name. In those cases, the models might simply reproduce generic patterns instead of generating original patterns tailored to a specific input image. To investigate this possibility, two experiments were designed. The first experiment evaluates the models’ reliance on generic patterns by swapping the source images for the first and second examples of each animal during the pattern accuracy assessment. If the model generates generic patterns based solely on the animal type rather than the specific visual characteristics of the input image, there should be no substantial difference in scoring when the pattern is compared to a different amigurumi version of the same animal. The second experiment tests the models’ ability to generate original patterns when provided with tasks lacking existing reference patterns. This is achieved by prompting the models with source images of non-existent or infrequently found amigurumi creatures. Such patterns are unlikely to be present in the models’ training data. Therefore, the quality and accuracy of the generated outputs in this context provide insight into the models’ ability to create new, coherent crochet patterns based on unfamiliar visual input.

4.3.1 Mismatch image experiment

To assess whether the generated patterns are genuinely derived from the provided source images or merely represent generic templates for a given animal, this experiment introduces a mismatch condition during the evaluation process. The original source image of a pattern is replaced with another amigurumi image of the same animal to compare the features between this new image and the existing pattern. The original, correctly paired data points are referred to as ‘matched’,

whereas the altered ones are classified as ‘mismatched’. The resulting scores for both matched and mismatched data pairs are presented in the boxplots in Figure 17. Across all score metrics for both models, GPT-4o and o1, a clear pattern emerges. Both the mean and median scores are consistently higher for matched data compared to mismatched counterparts. This trend is particularly evident in the color scores, which exhibit a substantial drop in performance under the mismatched condition. This indicates that the patterns are indeed generated based on the originally provided source images, as the colors can vary greatly across the images generated for the same animal (see Figure 29), possibly resulting in low color scores in mismatched cases.

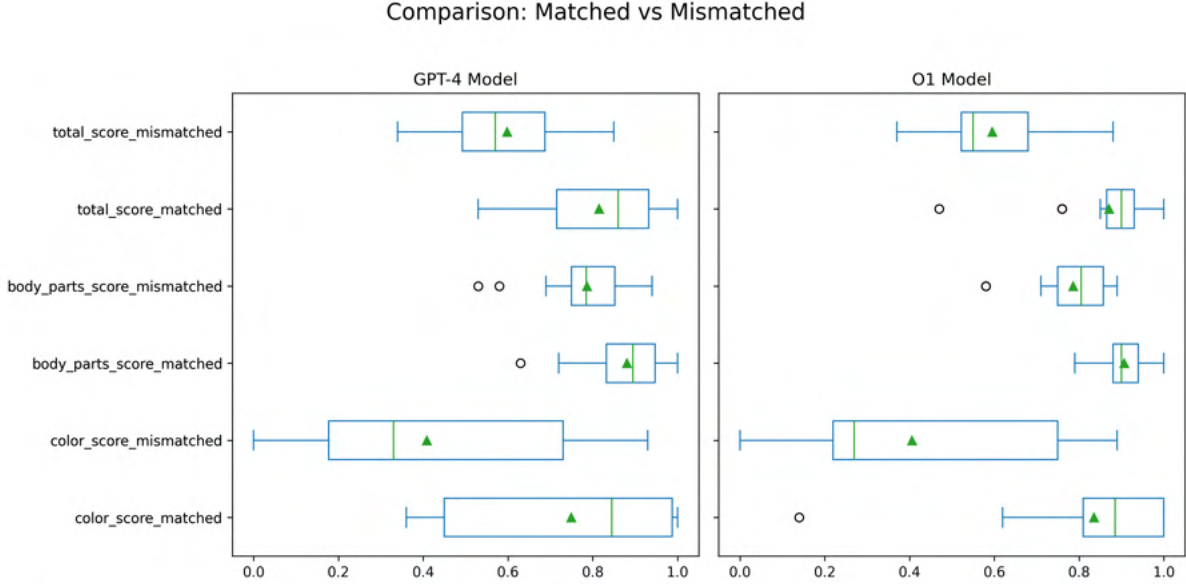


Figure 17: Boxplot of score distributions depending on model and comparison image. Green triangles depict the means, and the green lines the median.

To determine whether the observed differences between matched and mismatched scores are statistically significant, two-sided Wilcoxon signed-rank tests were conducted, appropriate for paired data that does not follow a normal distribution, as verified by the Q-Q plots (see Figures 18 and 19). The Wilcoxon signed-rank tests are performed with the Null-Hypothesis that there is no significant difference between the scores for matched pairs versus the scores for mismatched pairs, and the significance level of $\alpha = 0.05$, meaning that if the resulting p-values are below 0.05, the null hypothesis is rejected concluding that there is indeed a significant difference between the matched and mismatched scores. The statistical analysis reveals that for both models, the differences in color score, body-parts score, and total score between matched and mismatched pairs are all statistically significant (see Tables 5 and 6). These findings strongly indicate that the AI-generated patterns are not generic approximations but are instead meaningfully adapted to the specific characteristics of the source image.

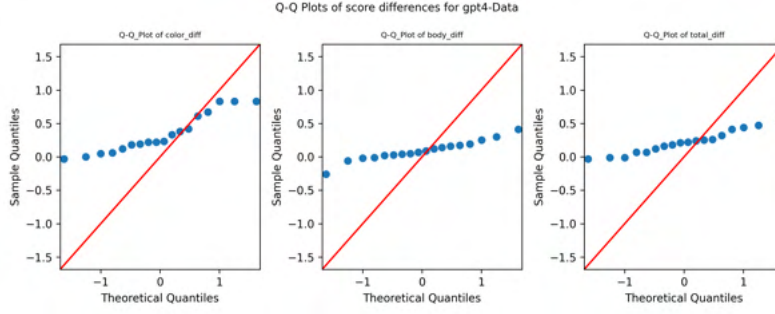


Figure 18: Q-Q plots for the score differences to check for normal distribution for GPT-4o data

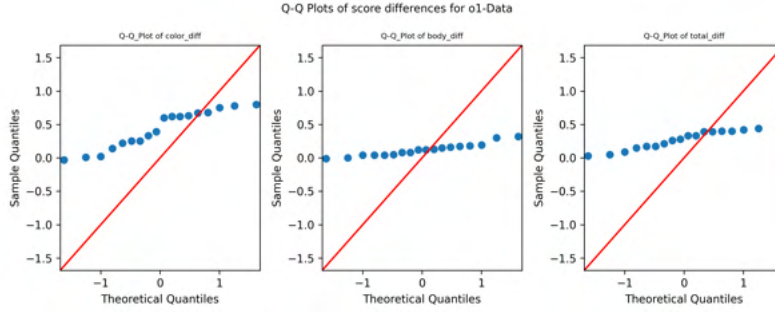


Figure 19: Q-Q plots for the score differences to check for normal distribution for o1 data

	color score diff	body part score diff	total diff
W	1.0	26.5	6.0
p-Value	0.0003	0.0102	0.0005
significance	significant	significant	significant

Table 5: Wilcoxon signed rank test with paired samples for matched and mismatched images for GPT-4o with significant p-value $\alpha < 0.5$

	color score diff	body part score diff	total diff
W	3.0	1.0	0.0
p-Value	0.0003	0.0003	0.0002
significance	significant	significant	significant

Table 6: Wilcoxon signed rank test with paired samples for matched and mismatched images for o1 with significant p-value $\alpha < 0.5$

4.3.2 Mythical creatures experiment

This experiment evaluates the capability of LLMs to generate original and plausible crochet patterns when provided with source images of unconventional or non-existent amigurumi designs. The

dataset for this test comprises four unique cases, including: a three-legged dog (see Figure 21), two fictional mythical creatures invented by the image generation model DALL-E 3 (see Figures 22 and 23), and an amigurumi representation of the coronavirus (see Figure 24). By using such atypical examples, the experiment seeks to assess whether the pattern generation process can produce coherent and realistic crochet instructions in the absence of corresponding training data.

The originality of the generated patterns is determined by two distinct aspects. First, the generative image model DALL-E 3 plays a key role in conceptual creativity as it is responsible for producing plausible amigurumi-style images based on textual prompts. Second, the pattern generation model (e.g., GPT-4o or o1) is responsible for interpreting the visual features of the generated image and translating them into crochet pattern instructions. This translation process becomes particularly challenging when there is no previous crochet pattern data on the depicted amigurumi in the source image, requiring the model to extrapolate instructions with a high degree of novelty.

Evaluation of the creative capacity of DALL-E 3 for this task yielded mixed results. In several instances, particularly with the invented mythical creatures, the model demonstrated strong generative capabilities, even including illustrative sketches in the background to support the design concept (see Figure 22). However, in cases requiring a deviation from standard anatomical features, such as the three-legged dog, the model failed to adhere to the prompt, generating an image of a standard four-legged dog instead (see Figure 20). This suggests a limitation in the model’s ability to override deeply embedded priors derived from its training data.



Figure 20: Amigurumi dog with only three legs generated by DALL-E-3

To further test the robustness of the pattern generation pipeline, an alternative image of a three-legged dog was produced using the newly released image generation capabilities of GPT-4o (i.e, GPT Image 1), which was occasionally able to meet the visual requirements. However, the subsequent pattern generation yielded inaccurate results. Both GPT-4o and o1 models produced patterns that included instructions for four legs. This outcome suggests limitations in the originality of the generated patterns. Notably, during the automatic feature extraction process, the image was incorrectly identified as depicting four legs, despite the presence of only three legs. This finding indicates that the source of the error may not lie solely in the pattern generation stage but may instead originate from inaccurate image-to-text processing, which supplies misleading component data to the pattern generation model.



Figure 21: Three-legged dog

When analyzing the AI-generated patterns for the first mythical creature (see Figure 22, there are several notable anomalies, particularly in the output produced by the GPT-4o model. At first glance, the pattern appears plausible, containing mainly the correct colors and body parts and hence receiving relatively high scores. However, a more in-depth examination of the patterns reveals a number of structural inconsistencies. For instance, the instructions for the horns create hollow cylinders that remain open on both ends and are not filled with stuffing, which deviates strongly from how a human would construct crochet horns. Furthermore, there is a mismatch in stitches in the final rows of the body and the tail, with the body having twice as many stitches, making a smooth connection of the body parts problematic. Additional issues include the complete omission of ears in the patterns and the oversimplification of the wing structure, lacking the originally depicted individual feathers. Similarly, the tail lacks the scale detailing which was depicted in the source image. Unfortunately, many of these inaccuracies, most notably the structural flaws in the horns, are not captured by the 2D representation, aside from the missing anatomical features (e.g., the ears).

In contrast, the pattern instruction generated by the o1 model demonstrates greater structural coherence. The pattern included logical details such as a suggestion to add a wire within the horns to preserve their curled shape. Unlike the GPT-4o pattern, the horns in this version are constructed as hollow cones, which are then lightly stuffed, resulting in a more accurate shape. The wings in the o1 pattern are constructed out of multiple feather segments of various sizes. However, the instructions for scaling these feathers lack clarity, as the pattern mentions that the size should be adapted but does not provide a specific method for modifying the base instructions. Additionally, the o1 pattern applies a popular crochet technique by recommending the use of eyelash yarn, which is known to create a more “fluffy” appearance, for the chest tuft. Overall, the comparative analysis of the two patterns suggests that the o1 pattern is more accurate to the original amigurumi. This also reflects in the observed pattern scores, where both the color and body parts scores are higher for the o1 pattern (see table 7). Nevertheless, the 2D visual representation of the o1 pattern does not exhibit a corresponding increase in quality.

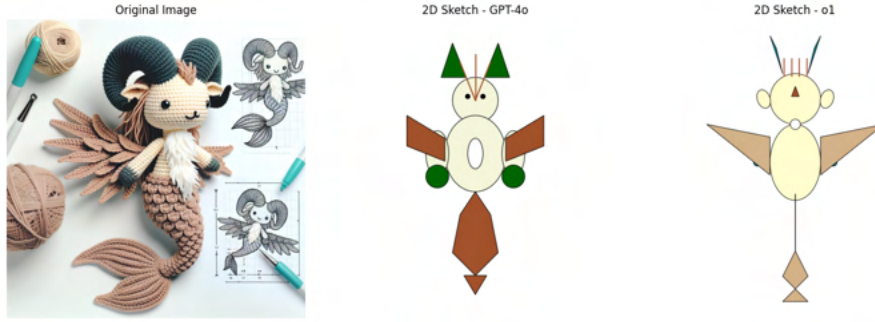


Figure 22: First mythical creature

For the second mythical creature, the patterns of both models appear generally plausible, with each containing most components and their corresponding shapes (see Figure 23). The pattern produced by GPT-4o includes most of the body parts shown in the original source image, which is also reflected in its high body parts score of 0.95 (see Table 7). The color score, on the other hand, is considerably lower. This is consistent with qualitative observations made on the pattern, for instance, the wrong shade of green being used for the ears or the use of brown for the hooves rather than the grayish-purple shown in the image. Surprisingly, for this creature, the model paid attention to details by including instructions for a fluffy mane, where it suggests attaching faux fur to achieve the desired appearance. One feature missing in the pattern instructions of GPT-4o is the spiral antenna at the top of the creature’s head. Its absence may be attributable to the features’ glowing appearance, which does not resemble realistic crochet.

In comparison, although the o1 model produced a pattern with a higher total score than GPT-4o (with an identical body-parts score) (see Table 7), it also contained some notable inaccuracies. Most prominently, the pattern included instructions for a dark green muzzle, which is not present in the source image. Nonetheless, there are, however, also some features from the original amigurumi which were mentioned by the o1 model but omitted in the GPT-4o pattern. Examples of this are the leaf detail at the end of the tail and the spiral antenna, for both of which the o1 pattern provides plausible instructions. Additionally, the pattern also suggests two options to add the fluffy mane detail by either using faux fur or by crocheting small teardrop-shaped pieces, which are then layered to create a leafy appearance. Overall, both models provide promising patterns for this unknown creature.



Figure 23: Second mythical creature

In the case of the coronavirus amigurumi, both GPT-4o and o1 generate plausible crochet patterns consisting of a spherical main body and spikes, which are attached in the crocheting process (see Figure 24). While both models provide instructions for creating individual spikes, the resulting spike shapes are likely to diverge from the reference image. Specifically, the spikes produced by the GPT-4o model are of a more ellipsoid shape, lacking the distinct stem evident in the source image. In contrast, the o1 pattern instructions include detailed steps to create the spikes, including a spherical shape at the top, followed by a narrow stem connecting the spike to the main body. However, during the actual crocheting process, the intended spherical component is not large enough to be visible, and hence, a shape closely resembling a rounded cone is created. Additionally, the pattern instructions from o1 include detailed instructions regarding the spatial arrangement of the spikes, mentioning general attachment points on the main body and the recommended distances between the spikes. Although the o1 pattern appears to be of higher quality in terms of component shapes and detailed descriptions, it received a lower score for the body parts evaluation metric (see table 7). Upon further inspection, this lower score is attributable to the inclusion of embroidered details in the pattern (e.g., a mouth and eyebrows) which were not present in the source image.

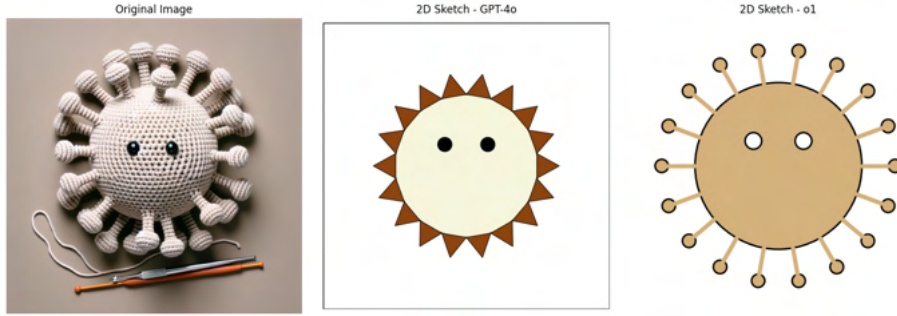


Figure 24: Coronavirus amigurumi

Model	Amigurumi	Body Parts Score	Color Score	Total Score
GPT-4o	Mythical Creature 1	0.77	0.65	0.71
	Mythical Creature 2	0.95	0.31	0.63
	Coronavirus	0.83	1.00	0.92
	Three-legged Dog	0.88	0.88	0.88
O1	Mythical Creature 1	0.89	0.91	0.90
	Mythical Creature 2	0.95	0.72	0.84
	Coronavirus	0.63	1.00	0.82
	Three-legged Dog	0.84	0.81	0.82

Table 7: Comparison of Scores for Unconventional Amigurumi by Model

4.4 Testing AI’s ability to adapt existing patterns

One approach to evaluating the extent to which LLMs comprehend the underlying logic of crochet patterns involves tasking the model with modifying specific components of an existing pattern. If a model possesses a functional understanding of crochet structure, it should be capable of adjusting

row and stitch counts in a coherent manner. In this study, both the standard GPT-4o model and the reasoning-enhanced models o1 were assessed on their ability to modify the size, length, and shape of components within amigurumi crochet patterns.

In a representative experiment, each model was provided with a full crochet pattern and instructed to enlarge the head of the amigurumi (i.e., the first gecko pattern generated by GPT-4o) (see Figure 25). Results indicate that both models successfully identified the relevant section of the pattern and modified it appropriately (see Figure 26). Specifically, the models introduced one or two additional rows of stitch increases, thereby expanding the total stitch count and circumference of the corresponding rows. To preserve the spherical shape of the head, the models symmetrically extend the pattern with corresponding decreased rows. Moreover, the models incorporated extra rounds of single crochet stitches in the middle, without which the resulting head would have taken a disc-like shape (oblate spheroid). Finally, the models adjusted the placement and spacing of the safety eyes to correspond to the increased dimensions, thereby maintaining relative positioning on the face.

```
### Head
Using light green yarn:
1. Magic ring, 6 sc [6]
2. Inc around [12]
3. (Sc, inc) around [18]
4. (2 sc, inc) around [24]
5. (3 sc, inc) around [30]
6. (4 sc, inc) around [36]
7. (5 sc, inc) around [42]
8-14. Sc around [42]
15. (5 sc, dec) around [36]
16. (4 sc, dec) around [30]
17. (3 sc, dec) around [24]
18. (2 sc, dec) around [18]; insert safety eyes between rows 12-13, approx. 8 sts apart
19. (Sc, dec) around [12]
20. Dec around [6]
- Fasten off and weave in the end.
```

Figure 25: Snipped of original pattern for the gecko head

```

### Adapted Head Pattern
Using light green yarn:
1. Magic ring, 6 sc [6]
2. Inc around [12]
3. (Sc, inc) around [18]
4. (2 sc, inc) around [24]
5. (3 sc, inc) around [30]
6. (4 sc, inc) around [36]
7. (5 sc, inc) around [42]
8. (6 sc, inc) around [48] // Increased size
9. (7 sc, inc) around [54] // Increased size
10-18. Sc around [54]
// Extended the number of rows to maintain round shape
19. (7 sc, dec) around [48]
20. (6 sc, dec) around [42]
21. (5 sc, dec) around [36]
22. (4 sc, dec) around [30]
23. (3 sc, dec) around [24];
insert safety eyes between rows 16-17, approximately 10-12 sts apart
24. (2 sc, dec) around [18]
25. (Sc, dec) around [12]
26. Dec around [6]
- Fasten off and weave in the end.

### Notes:
- Adjust the placement of the safety eyes
to align with the increased head size.
- Continue to stuff the head as
you go to maintain the shape and ensure it remains firm.

```

ADAPTED HEAD (LARGER)

Use light green yarn:

```

1) Magic ring, 6 sc [6]
2) Inc around [12] (work 2 sc in each stitch)
3) (Sc, inc) around [18]
4) (2 sc, inc) around [24]
5) (3 sc, inc) around [30]
6) (4 sc, inc) around [36]
7) (5 sc, inc) around [42]
8) (6 sc, inc) around [48]
9-16) Sc around [48] (8 rounds of single crochet)

```

Begin decreasing:

```

17) (6 sc, dec) around [42]
18) (5 sc, dec) around [36]
19) (4 sc, dec) around [30]
20) (3 sc, dec) around [24]
21) (2 sc, dec) around [18]
22) (Sc, dec) around [12]
23) Dec around [6]

```

- Fasten off and weave in the end.

NOTE: Because the head is larger, consider placing safety eyes slightly farther apart than before—about 10-12 stitches apart. A good spot is between rounds 13 and 14, or wherever looks most natural based on your preference.

Figure 26: Generated gecko head pattern modified by gpt-4o (left) and o1 (right)

Another experiment to evaluate LLMs’ pattern modification capabilities involves modifying the length of body parts rather than their overall size. This type of transformation required a different crochet strategy, primarily through the addition of single crochet rows rather than increases in stitch count. Figure 28 presents the results of an experiment in which the models were provided with an amigurumi pattern and tasked with generating instructions for longer arms. In the original sample pattern, the arms and legs were described jointly, as they shared identical shapes (see Figure 27). To perform the modification successfully, the language models had to restructure the instruction, separating the arm and leg components into distinct sections.

The GPT-4o model approached this by explicitly stating that it would extend the arm length by creating a modified version of the leg instructions. It accurately identified a suitable insertion point for additional rows, at the upper arm, without significantly altering the shape. Furthermore, it offered flexibility by suggesting variable row counts depending on the desired arm length. Similarly, the o1 model restructured the pattern by independently defining the legs and arms. For the arm instructions, it added two additional rows by default and also included guidance on how further extensions could be achieved by adding additional rows. Both models demonstrate an impressive ability to reason about structural length modification within crochet patterns.

```

### Legs (Make 4)
Start with dark green yarn:
1. Magic ring, 6 sc [6]
2. Inc around [12]
3-5. Sc around [12]
6. Change to light green yarn, (Sc, dec) around [8]
- Stuff lightly.
7-8. Sc around [8]
- Fasten off, leaving a long tail for sewing. Attach them to the base of the body, two for arms and two for legs.

```

Figure 27: Snipped of original pattern for the gecko arms/legs

To make the arms of your lizard amigurumi longer, you can simply add more rounds to the leg pattern you're currently using for the arms. Here's how you can adjust the pattern for longer arms:

Adjusted Arms Pattern (Longer Version)

```

1. **Magic ring, 6 sc** [6]
2. **Inc around** [12]
3-5. **Sc around** [12]
4. **Change to light green yarn, (sc, dec) around** [8]
5. **Continue with light green yarn, Sc around** [8]
   | for as many additional rounds as needed for your desired arm length.
   | - To extend just a bit, you might add 2-3 rounds here.
   | For even longer arms, add up to 6-8 more rounds based on your preference.
6. **Fasten off, leaving a long tail for sewing**.
Attach them to the base of the body as arms.

```

Integration

1. After crafting the longer arms, ensure they are proportionally and symmetrically attached to the body.
2. You can also adjust their curve or pose while stuffing lightly to give them more realistic positioning if desired.

By adding the extra rounds above, you can customize the arm length to suit your preference. Enjoy crafting your unique lizard amigurumi!

Below is a suggested modification to make the arms longer while keeping the legs as originally designed. Everything else in the pattern can remain the same unless you wish to make further changes.

ADAPTED ARMS (MAKE 2)

Use the same yarn colors as before for a cohesive look. If you'd like your arms to match exactly, start with dark green, then switching to light green as in the original pattern.

```

1) Magic ring, 6 sc [6]
2) Inc around [12]
3) Sc around [12]
4) Sc around [12]
5) Sc around [12]

6) Change to light green yarn, (sc, dec) around [8]
   | - This narrows the arm to 8 stitches. Lightly stuff as you go.

• To make the arms longer, simply work more rows in this narrower section:

7) Sc around [8]
8) Sc around [8]
9) Sc around [8]
10) Sc around [8]
   | - You can add or subtract rows here to make the arms as long or short as you want.

• When you are satisfied with the length, fasten off, leaving a long tail for sewing to the body.

```

LEGS (MAKE 2) [Unchanged]

If you want to keep the original leg length, follow the original pattern for the other two limbs:

```

1) Magic ring, 6 sc [6]
2) Inc around [12]
3-5) Sc around [12]
6) Change to light green yarn, (sc, dec) around [8]
   | - Lightly stuff.
7-8) Sc around [8]
   | - Fasten off, leaving a long tail for sewing.

```

Figure 28: Generated gecko arms pattern modified by gpt-4o (left) and o1 (right)

5 Conclusions

5.1 Results

The purpose of this study was to investigate the generative capabilities of LLMs in the context of amigurumi crochet pattern design. Through a multi-step pipeline involving image generation, pattern creation, and automated evaluation, it was possible to assess how well OpenAI's GPT-4o and o1 models translate generated crochet images into structured, human-readable crochet instructions. The results of the performed experiments testing originality suggest that both models perform reasonably well in creating patterns accurately based on the provided input, including their vital components and their corresponding colors. The conducted experiments indicate that the pattern generation procedure does not rely solely on memorized generic patterns but can generate original and specialized patterns directly linked to the source image. The pattern adaptability tests show that both GPT-4o and o1 can make context-aware modifications since both models were able to accurately adapt patterns according to specified requirements.

In practice, the models did not always show perfect understanding of the underlying crochet logic following patterns. Occasionally, instructions would include incorrect stitch counts in a row or wrongly completed color switches, making the pattern less plausible to human readers and distorting the final product. This indicates that although the AI-based crochet instruction generation process appears relatively effective, its reliability and quality can still be improved. Furthermore, the investigation on possible visualization techniques concluded that 2D representations are generally effective in depicting the presence and color of amigurumi components derived from the pattern. However, the 2D visualizations demonstrated limitations in their ability to accurately depict the shape and placement of the individual components, especially in the case of more complex amigurumi figures. The 3D visualizations proved largely unreliable, primarily due to a high rate of rendering errors and structural inaccuracies. Overall, the study shows promising results in crochet pattern generation using LLMs, which might further improve with future advancements in the field of AI.

5.2 Limitations

There are some limitations to this work which were previously addressed in this research, reducing the overall scalability of the project. One significant limitation is the reliance on accurate feature extraction for scoring. Due to the high variability and complex nature of patterns, it is challenging to construct an automatic, reliable feature extraction method. Consequently, feature extraction still occasionally requires manual corrections and human supervision. The scoring metric, while being systematic, does not fully capture pattern quality, as it fails to evaluate the composition and accuracy of created shapes in the instructions. Moreover, the small dataset size and subjective difficulty labels limit the extent to which the findings can be generalized, particularly when analyzing the correlation between AI’s performance and the corresponding difficulty levels of the amigurumi. A further limitation, particularly in the process of analyzing the effect of reasoning models on pattern generation, is that it is not possible to access the reasoning process performed during the generation of a response in the OpenAI API. A large limitation in the current creation of 3D representations is the lack of a Model Context Protocol (MCP) enabling LLMs to interact directly with the Blender environment. Through such an MPC the AI would be able to access the 3D environment and iteratively adapt its generated model, which could greatly improve the resulting 3D amigurumi models.

5.3 Future work

There are many options for future work to extend and refine the findings from this research. For example, OpenAI’s recently introduced image generation model is now integrated into GPT-4o (GPT Image 1), allowing it to use greater information resources (i.e., the versatile capabilities emerging from GPT-4o’s training) to analyze a prompt and generate a coherent image. This provides opportunities to apply this new image generation to visualize the resulting amigurumi, additionally simplifying comparison to the source image. By directly comparing these renderings to the source image, one could create a feedback loop that iteratively improves the pattern based on the generated representation, leading to a self-improving pattern creation pipeline. These adaptations might allow for a largely automated pattern evaluation framework, enabling the expansion of the dataset and repetition of experiments and significance tests, thus further strengthening the reliability of the observed results. Furthermore, these new capabilities would enable larger-scale experiments

than were feasible in this study. Specifically, it would pave the way for a wide array of automated “adaptation experiments”, adapting various aspects of patterns across many animals. This would allow for an extensive analysis of AI’s logic comprehension abilities in the area of crochet patterns.

Another potential extension of the current work would be to integrate a co-creative interface into the existing pattern generation pipeline. While full automation remains challenging, incorporating a human-AI collaboration framework could support amigurumi designers in their creative process. This enables the use of AI as a tool in crochet to create plausible pattern instructions. The interactive loop allows designers to iteratively refine specific pattern components by providing targeted feedback or modification instructions to the AI.

Further useful expansions of the existing work might involve evaluating the crochet pattern generation pipeline using a broader range of existing popular LLMs, such as Gemini, Claude, or DeepSeek. This would allow a systematic comparison of different models’ creative capabilities and their ability to produce more abstract program-like structures like crochet instructions.

An additional direction for future research would be to assess the performance of fine-tuned LLMs, trained on existing crochet data in comparison to general-purpose models. Such a comparison could offer insights into the trade-offs between broad linguistic competence and domain-adapted precision in creative tasks. As an extension of this research, it might also be interesting to investigate whether AI models can be fine-tuned to produce valid and plausible pattern instructions that can be visualized by crochet rendering software such as CrochetPARADE. To investigate this, it would first be essential to collect an extensive database of amigurumi patterns written in the formal syntax required by the software to train the model on. If successful, this approach could enable an efficient and detailed visualization process, which might easily be extended into a co-creative amigurumi designing environment. Furthermore, the methodology could be generalized to accommodate other crochet forms beyond amigurumi, thereby broadening its applicability within computational craft design.

References

- [Bod09] Margaret A Boden. Computer models of creativity. *Ai Magazine*, 30(3):23–23, 2009.
- [ÇBG17] Özgüç Bertuğ Çapınaman, Cemal Koray Bingöl, and Benay Gürsoy. Computing stitches and crocheting geometry. In *Computer-Aided Architectural Design. Future Trajectories: 17th International Conference, CAAD Futures 2017, Istanbul, Turkey, July 12–14, 2017, Selected Papers 17*, pages 289–305. Springer, 2017.
- [EPIBC22] Michal Edelstein, Hila Peleg, Shachar Itzhaky, and Mirela Ben-Chen. Amigo: computational design of amigurumi crochet patterns. In *Proceedings of the 7th Annual ACM Symposium on Computational Fabrication*, pages 1–11, 2022.
- [EY24] P. Ezhilmurugan and E. Yashavini. Exploring the impact of artificial intelligence in the visual arts: A comprehensive study. *ShodhKosh: Journal of Visual and Performing Arts*, 5(ICITAICT):74–84, 2024. International Conference on Innovative Techniques in Artificial Intelligence and Communication Technologies.
- [Khe23] Kheyo. Exploring the connection: Is crochet like coding?, 2023. Accessed: 2025-05-27.
- [KW24] Jiahui Zhang Shiqi Wang Wanying Zhang Kathy Wu, Yanfeiyun Wu. Yarn master: Ai-powered crochet/knit pattern instruction generator. https://github.com/kathywu1201/AC215_YarnMaster, 2024. GitHub repository.
- [Lan19] Ganaele Langlois. Distributed intelligence: Silk weaving and the jacquard mechanism. *Canadian Journal of Communication*, 44(4):555–566, 2019.
- [MXF⁺23] Suvir Mirchandani, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. Large language models as general pattern machines. *arXiv preprint arXiv:2307.04721*, 2023.
- [Ope24] OpenAI. Learning to reason with language models, 2024. Accessed: 2025-05-29.
- [Pap25] Donald Papp. Mcp blender addon lets ai take the wheel and wield the tools, May 2025. Accessed: 2025-05-23.
- [Run25] Mark A Runco. Updating the standard definition of creativity to account for the artificial creativity of ai. *Creativity Research Journal*, 37(1):1–5, 2025.
- [SGC⁺20] Christoph Salge, Michael Cerny Green, Rodrigo Canaan, Filip Skwarski, Rafael Fritsch, Adrian Brightmoore, Shaofang Ye, Changxing Cao, and Julian Togelius. The ai settlement generation challenge in minecraft: First year report. *KI-Künstliche Intelligenz*, 34:19–31, 2020.
- [SLRH21] Klara Seitz, Jens Lincke, Patrick Rein, and Robert Hirschfeld. *Language and tool support for 3D crochet patterns: virtual crochet with a graph structure*, volume 137. Universitätsverlag Potsdam, 2021.
- [Tas25] Svetlin Tassev. Crochetparade: Crochet pattern renderer, analyzer, and debugger, 2025. Accessed: 2025-05-23.

- [The19] The Citizens’ Voice. In this physics lab, ‘knitting is coding’ and yarn is programmable, 2019. Accessed: 2025-05-30.
- [WHL24] Roosa Wingström, Johanna Hautala, and Riina Lundman. Redefining creativity in the era of ai? perspectives of computer scientists and new media artists. *Creativity Research Journal*, 36(2):177–193, 2024.
- [Wu24] Kathy Wu. Ai-powered crochet/knit pattern instruction generator. <https://medium.com/institute-for-applied-computational-science/ai-powered-crochet-knit-pattern-instruction-generator-7111786413b4>, December 2024. Medium, Institute for Applied Computational Science.

6 Appendix

The complete code can be found in this Github repository: https://github.com/catsmey/ai_amigurumi_crochet_thesis

Task	System prompt	Prompt	Model Key
Source image generation	-	”Create an image of a simple crochet {animal}. The image has to look like an amigurumi that can be crocheted in real life.”	dall-e-3
Generate pattern	”You are a knowledgeable crochet pattern amigurumi generator. You analyze a given crochet amigurumi image and write a detailed crochet pattern on how to make it. You can always provide a pattern by converting the image into a text description and then creating a fitting pattern for it.”	”Analyze the amigurumi image in detail and generate a complete, row-by-row crochet pattern. Include stitch counts, exact placement of color changes, points in the patterns where a component needs to be filled with stuffing, and detailed connection points for all parts. Ensure that the pattern follows standard crochet notation (sc, inc, dec, etc.). Provide a materials list and finishing instructions.” + <Source Image >	gpt-4o or o1

Task	System prompt	Prompt	Model Key
2D visu- alization generation	"You are a Python code generator. When a pattern is provided, you will generate Python code using Matplotlib to create a 2D image of the amigurumi described in the pattern. The image should contain all components, with borders to ensure visibility, and should be saved as a PNG in the provided directory. Mention and clarify the code by adding comments mentioning all the individual components. Your response should only include the Python code, without any additional text, comments, or explanation. The pattern should be processed as described, and if no pattern is provided, return an empty 2D representation."	"Pattern: {pattern}. *end pattern*. Use Python, Matplotlib, and math to generate code to create a 2d image of the resulting amigurumi created by this crochet pattern. Include all components and connect them as mentioned in the pattern. Make sure to include a small border around each component to make sure they don't disappear in the background if they have the same color. Save the final image as a png in the existing directory (no need to import os) '{file_location}'. Don't include any text in your response so it can be run immediately! Only give code in your response. Don't say "python at the beginning. If no pattern was given, return an empty 2d representation."	o3-mini
3D visu- alization generation	-	"Use python blender to generate code to create a 3d model of the resulting amigurumi created by this crochet pattern. Including colors. Don't include any text in your response so it can be run immediately! Only give code in your response. Don't say "python at the beginning of your response. Pattern: {pattern}"	o3-mini
Mythical creature source image generation	-	"Invent a mythical creature and generate an image of a crochet amigurumi of that creature. The image has to look like an amigurumi that can be crocheted in real life."	dall-e-3
Adapt exist- ing pattern	-	"Adapt the crochet pattern to: {adaptation_component}. Here is the original pattern: {pattern}"	gpt-4o or o1

Task	System prompt	Prompt	Model Key
Feature extraction from image	"You are a Python code generator. Given a crochet amigurumi image, identify and list all the body parts and features of the amigurumi. Return a dictionary with the body part as the key, and a list containing: 1. The number of times this body part appears in the pattern (if specified). For components like stripes, dots, fluffy texture or features like whiskers, or blush on the cheeks, return a True (correct python format) value if they are mentioned. 2. A list of colors associated with that body part. Colors in the dictionary must be: A single basic color name (e.g., 'red', 'blue', 'green'). No other descriptor prefixes are allowed. Do not include additional adjectives or complex descriptions. Only use lowercase letters for the whole dictionary. Provide only the dictionary without any additional text or explanations. Also don't write "python."	<Image>	o1

Task	System prompt	Prompt	Model Key
Feature extraction from pattern	"You are a Python code generator. Given a crochet pattern, identify and list all the body parts and features mentioned in the pattern. Return a dictionary with the body part as the key, and a list containing: 1. The number of times this body part appears in the pattern (if specified). For components like stripes, dots, fluffy texture or features like whiskers, or blush on the cheeks, return a True value if they are mentioned. If no pattern is given, return an empty dictionary. 2. A list of colors associated with that body part. Colors in the dictionary must be: A single basic color name (e.g., 'red', 'blue', 'green'). No other descriptor prefixes are allowed. Do not include additional adjectives or complex descriptions. Only use lowercase letters for the whole dictionary. Provide only the dictionary without any additional text or explanations. Also don't write "python."	Pattern: <Pattern>	o1

Task	System prompt	Prompt	Model Key
Feature extraction from 2D visualization	"You are a Python code generator. Given a code for a 2d image, identify and list all the body parts and features mentioned in the pattern. Return a dictionary with the body part as the key, and a list containing: 1. The number of times this body part appears (if specified). For components like stripes, dots, fluffy texture or features like whiskers, or blush on the cheeks, return a True value if they are mentioned. If no code is given, return an empty dictionary. 2. A list of colors associated with that body part. Colors in the dictionary must be: A single basic color name (e.g., 'red', 'blue', 'green'). Do not include additional adjectives or complex descriptions. Only use lowercase letters for the whole dictionary. Provide only the dictionary without any additional text or explanations. Also don't write "python."	2d representation code: <matplotlib code>	o1

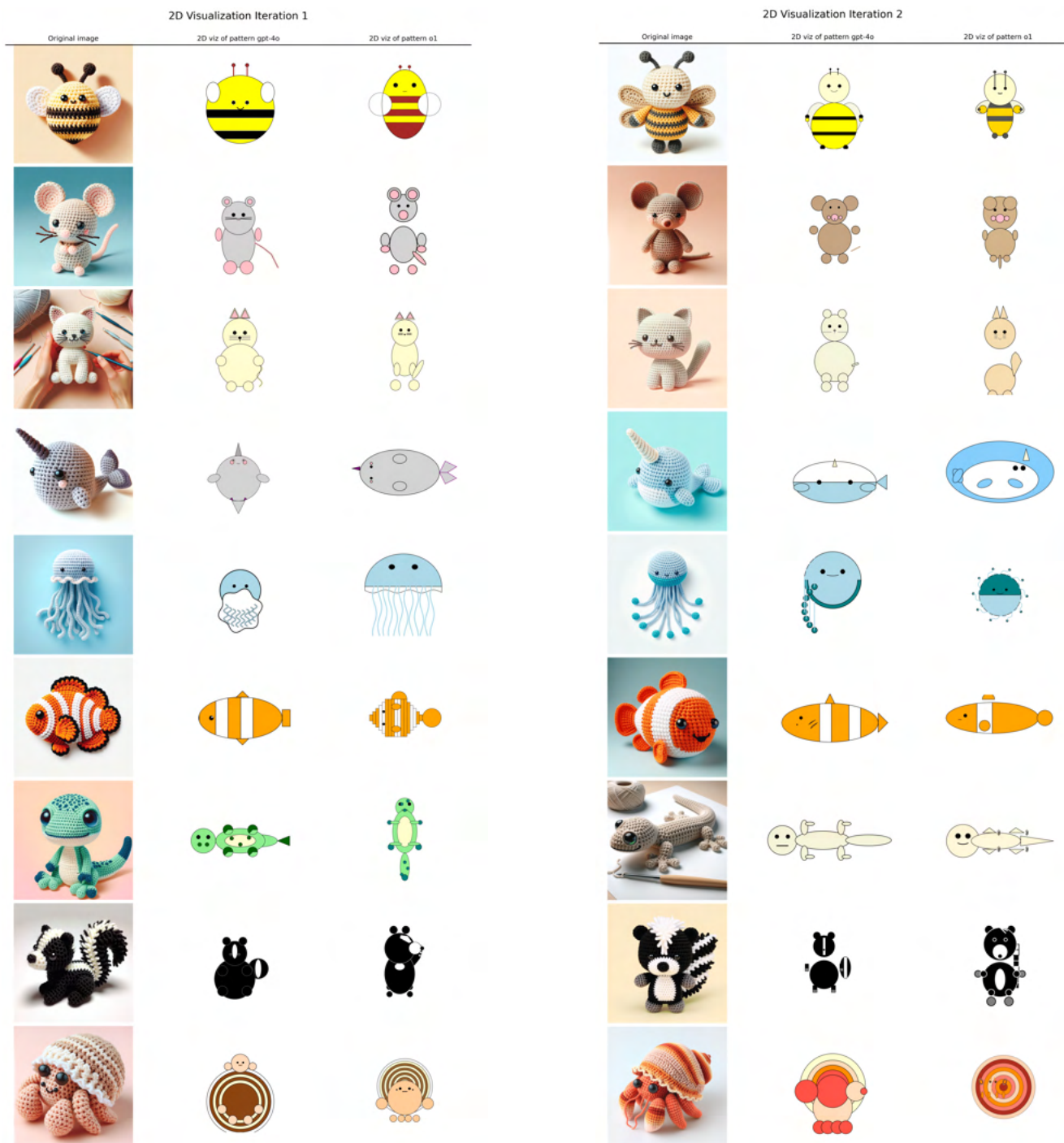


Figure 29: Overview of the 2D visualizations of the generated patterns



Figure 30: Overview of the 3D visualizations of the generated patterns