Amazon Titan Image Generator
Prompt Engineering Best Practices

Table of Contents

Introduction

Welcome to the prompt engineering guide for Titan Image Generator models, Amazon’s Foundation Models for image generation!

Amazon Titan is available as part of Bedrock, Amazon’s service for working with foundation models (FMs).

Prompt engineering refers to the common practice of optimizing the textual input to FMs to obtain desired responses. Prompting allows image FMs to perform a wide variety of tasks including image generation from a text instruction, as well as different ways to perform image editing for existing images through text instructions. The quality of the prompts you give to Amazon Titan Image Generator models can have a large effect on the quality of its responses. Our aim is to provide you with all of the necessary information to get started with prompt engineering as well as the tools to quickly find the best possible prompt format for your use case when using Amazon Titan models for generating images.

Whether you’re a beginner in the world of generative AI, or an expert with previous experience, this guide will help you optimize your prompts with Amazon Titan Image Generator models.

As the philosopher Ludwig Wittgenstein wrote in the *Tractatus Logico-Philosophicus*, “*Whereof one cannot speak, thereof one must be silent.*”

Ludwig Wittgenstein, *Tractatus Logico-Philosophicus,* Translated by CK Ogden
Proposition 7, page 90

Now, with Amazon Titan, let us speak.

What is an image model prompt?

An image model prompt is a text description or instruction that is provided as input to a text-to-image or to an image-and-text-to-image foundation model, to guide it in generating a corresponding image output. They describe the desired image to be generated in words, allowing the model to interpret that text and create a matching visual representation.

The prompts for image models can vary from being very specific, e.g.: "a photo of a red flower in a blue vase on a table", to more abstract, e.g.: "a surreal dreamscape with swirling colors". They leverage the knowledge the foundation model has gained from its training data to "imagine" what an image matching the prompt might look like.

Prompts constrain the output space for the model and make the image generation more controllable compared to generating images from scratch. Effective prompts tend to be detailed but not overly long, providing key visual features, styles, emotions or other descriptive elements. The same foundation model can produce very different images based on the prompt provided as input.

Prompt engineering, or iterating on prompts to get better results, is an important part of leveraging text-to-image models effectively. The text prompt acts as a guide for the model to create a corresponding image, allowing users to steer the image generation process.

Components of a prompt for image generation FMs

In general, there are a few key components that make up a prompt for text-to-image or image-and-text-to-image foundation models:

- Text description: This provides a textual description of the desired image. It can include details about the subject, style, composition, etc. The text prompt encodes what the model should generate.

- Image conditioning: An existing image can be provided as a starting point or reference that the model should incorporate. This conditions the model and provides additional guidance.

- Guidance tokens: Special tokens can be added to provide higher-level guidance like "photorealistic", "cinematic lighting" etc. These act like keywords for the model.

- Parameter selection: Things like image resolution, sampling method, number of iterations, etc. can be specified to control the final output.

In summary, the prompt contains text descriptions, image references, style keywords, parameter selections and special tokens to guide the foundation model towards generating the desired image from text, or image and text, inputs.

What is prompt engineering for image models?

Prompt engineering refers to the process of carefully crafting the text prompts provided to text-to-image or image-and-text-to-image models, in order to generate the desired images. The prompts act as instructions that guide the AI system on what type of image to generate.

Some key aspects of prompt engineering include:

- Word choice - Selecting descriptive words and phrases that capture key visual details. Using more vivid, concrete language often works better than vague descriptions.

- Style and tone - The prompt can be conversational, poetic, humorous, etc. to influence the mood/style of the generated image.

- Structure - Logically ordering prompt elements and using punctuation to indicate relationships. For example, commas to separate independent clauses or colons to lead into a description.

- Guiding details - Adding specific details to narrow down the scope and guide the image generation, like object shapes, colors, lighting, background scene, etc.

- Negative prompts - Specifying what should NOT be included in the image to avoid unwanted elements.

- Iterative refinement - Progressively editing the prompt through multiple cycles based on the results.

- Drawing inspiration from examples - Leveraging descriptions of existing images to inform prompt construction.

The goal is to find the right prompt wording to produce the intended visuals from the models. As it happens with Large Langue Models (LLMs) for text generation, Prompt Engineering is also considered a key skill in text-to-image generation.

Amazon Titan Image Generator Prompts

Image Generation

For image generation with Titan, most of the same guidelines of other text-to-image models apply. An exception could be quoting text within the prompts, where Titan works better when provided with double quotes, e.g.: *An image of a boy holding a sign that says "success"* instead of: *An image of a boy holding a sign that says 'success'*.

In general, consider the following recommendations:

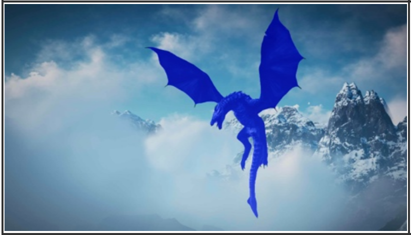
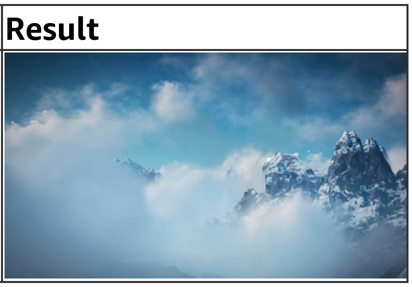
- Start the prompts with the subject, e.g.: "An image of a …"
- Provide as much details as possible, if relevant include the desired: medium, color, lightning, remarks, adjectives, quality, and style
- Use specific prompts and, if applicable consider using negative prompts
- Rely on the model parameters

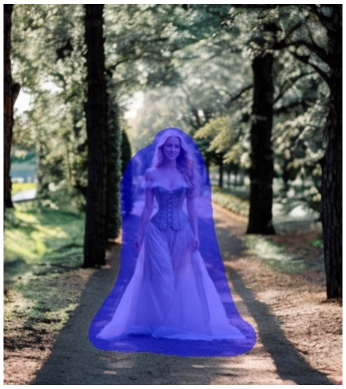
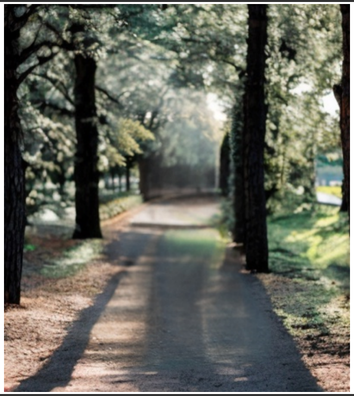
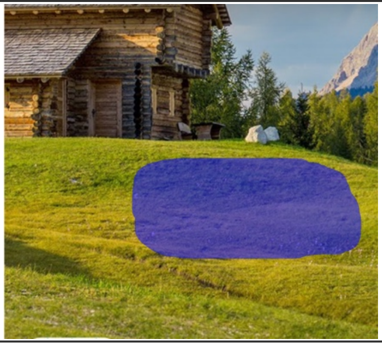
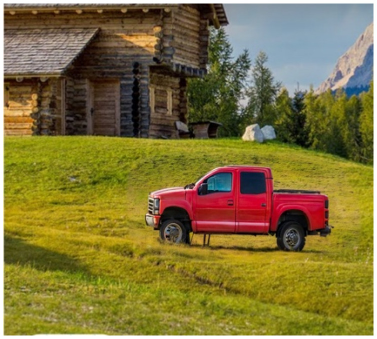
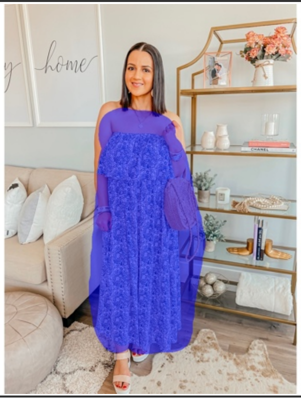
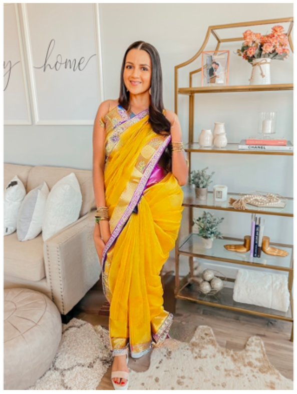
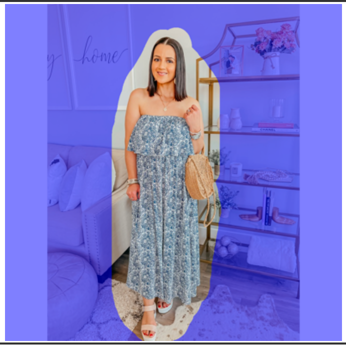
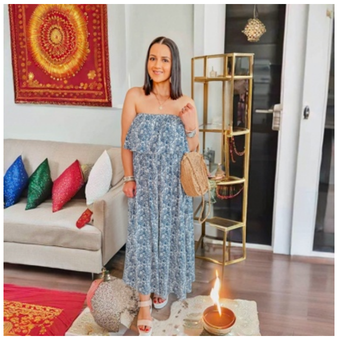
Inpainting & Outpainting

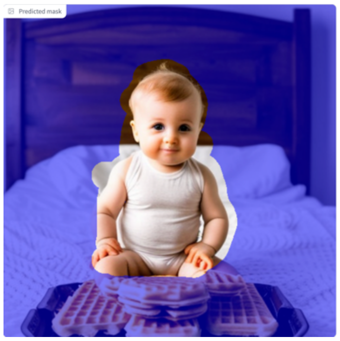
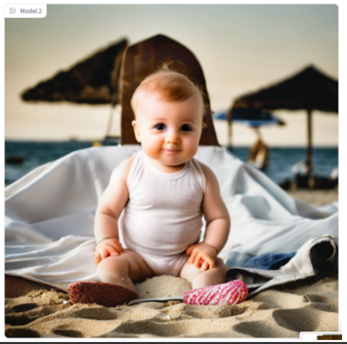
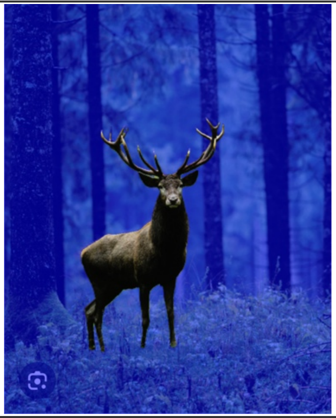
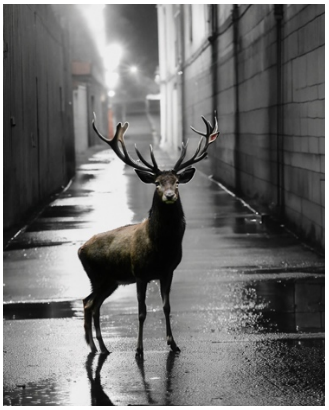
Both inpainting and outpainting are processes that take: an image, a "text prompt", and a segmentation mask as inputs. The segmentation mask is optional; alternatively, you can also input a "mask text prompt" and use Titan' segmentation algorithms to estimate the mask based on that prompt.

In general, Inpainting reconstructs the region within the mask, while Outpainting generates new pixels that seamlessly extend the mask region.

Here some examples:

Original image	API	Text Prompt	Result
	Inpainting Removal	""	

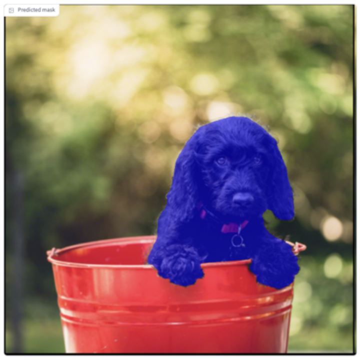
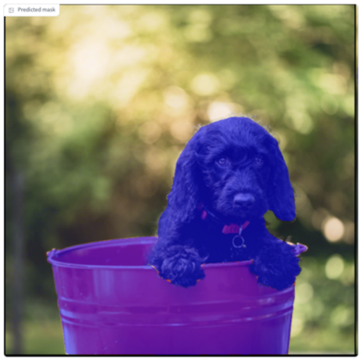
	Inpainting Removal	""	
	Inpainting Default	“red truck parked"	
	Inpainting Default	“yellow saree"	
	Outpainting Default	“Diwali decorations"	

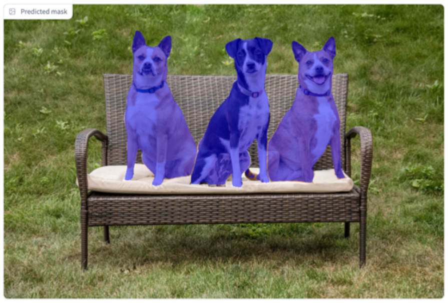
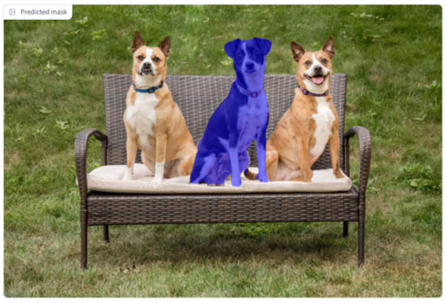
	Outpainting to extend image boundaries	“on a beach"	
	Outpainting to generate different backgrounds	“Deer in a wet nighttime alley"	

We will now see some recommendations of each of the tasks.

Mask Prompt

This prompt is optional, and only needed when you don’t provide a segmentation mask for the image. The prompt itself should specify what object(s) to segment. It has to be precise and detailed about those, as that information is used to guide the segmentation. For example, in the images below you have to specify unique attributes/features of the center dog to drive the segmentation algorithm to focus on that dog only.

	Original image	Result
Example1		

Mask Text prompt	Dog	Dog in a bucket
Example2	 	
Mask Text prompt	Dog	Black dog

Prompt for Inpainting Object Removal

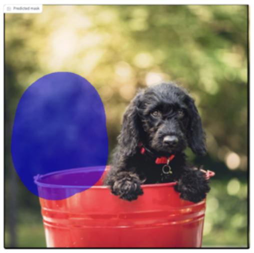
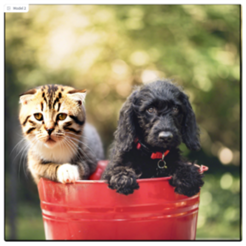
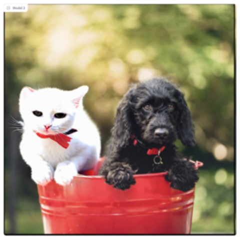
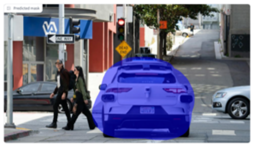
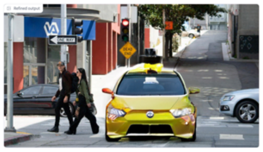
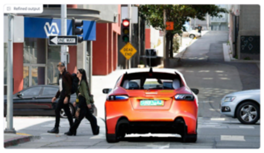
The text prompt should be empty.

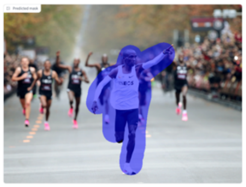
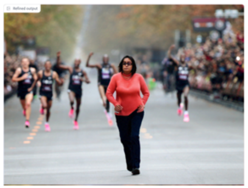
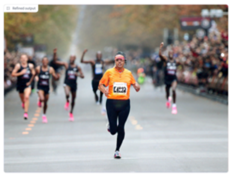
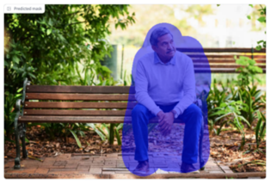
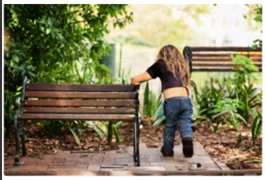
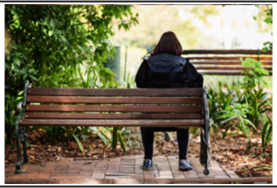
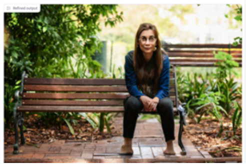
Prompt for Inpainting Object Replacement

It should be precise about:

- what to reconstruct inside the mask region, including its unique features/attributes (e.g., color, viewpoint, etc.). See Examples 1-4 below
- (optional, but be very helpful) some details about the context/background of the object, to help inpainting improve the reconstruction (more realistic). See Example 3-4 below

For example:

	Original mask (customer provided)	**Example Output**	**Example Output**	**Example Output**
Example1				-
Mask Text prompt	-	A cat	A white cat	-
Example2				-

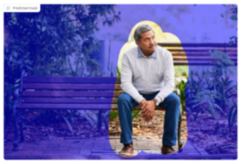
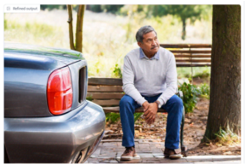
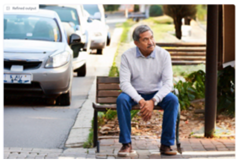
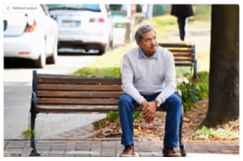
Mask Text prompt	-	A car	A car from the back	-
Example3				-
Mask Text prompt	-	A person	A person running a marathon	-
Example4				
Mask Text prompt	-	A person	A person sitting on a bench	A person sitting on a bench facing the camera

Prompt for Outpainting

It should follow a similar level of details as Inpainting:

- A detailed description of the background to outpaint. The more details are provided, the more the image will align with them. See examples below.
- (optional) sometimes it is useful to describe the whole scene, including the object inside the mask that we want to outpaint. While this information is often redundant, e.g.: in examples 1 and 2 below, it does not hurt the outpainting outcome. Also, it occasionally helps rendering some hard cases that the model cannot handle well, e.g.: in examples 3 and 4 below.

For example:

	Original mask (customer provided)	**Example Output**	**Example Output**	**Example Output**
Example1				

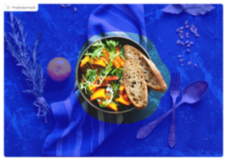
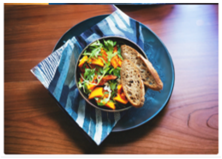
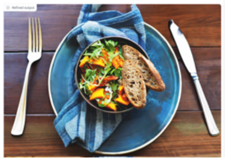
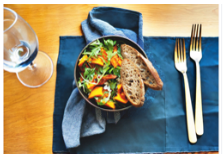
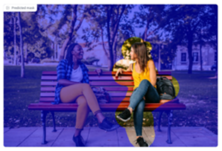
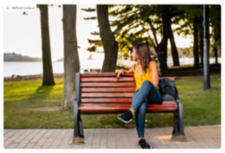
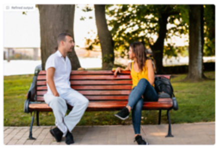
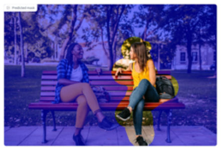
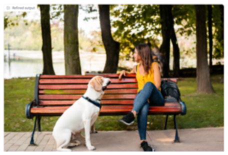
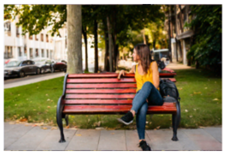
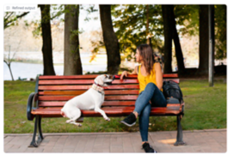
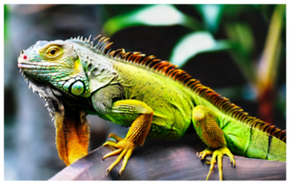
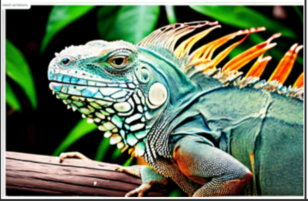
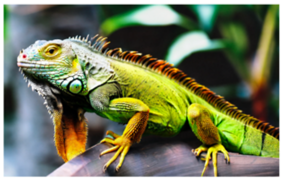
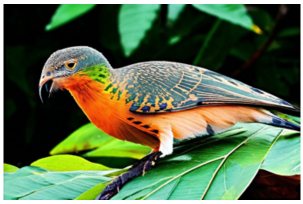
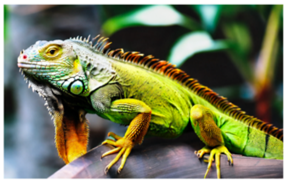
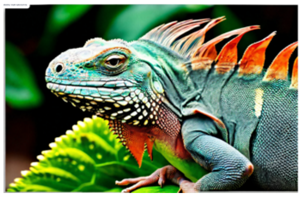
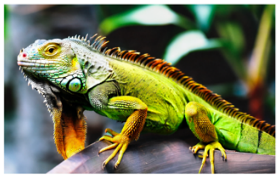
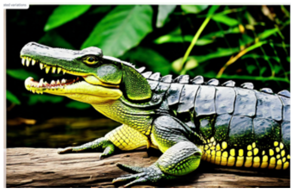
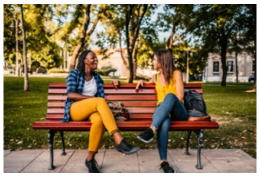
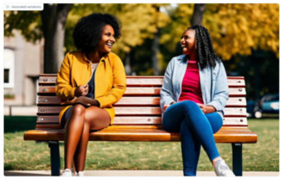
Mask Text prompt	-	Cars	A street with cars	A man sitting on a bench with a street with cars behind him
Example2				
Mask Text prompt	-	A wooden table	A wooden table with cutlery	A wooden table with cutlery and a glass
Example3				
Mask Text prompt	-	A person *(somehow the generation doesn't work)*	A person talking to another person *(more details fix the generation)*	
Example4				
Mask Text prompt	-	A dog	A dog sitting on a bench *(somehow the generation doesn't work)*	A dog sitting on a bench next to a woman *(more details fix the generation)*

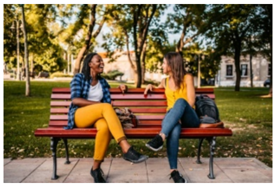
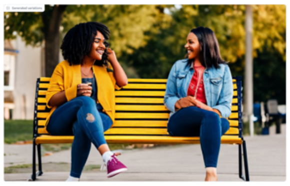
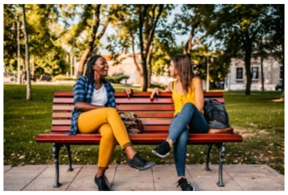
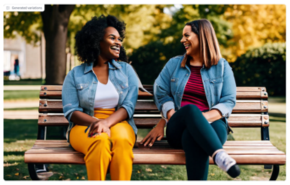
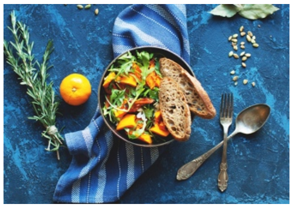
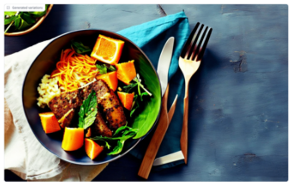
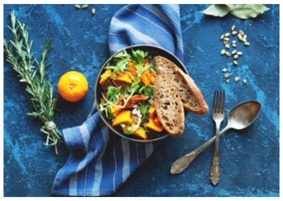
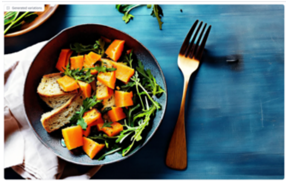
Image Variations

Image Variation takes an image and optionally a text prompt as input, and generates a new image that preserves the content of the input image but varies it a little.

The text prompt should describe the *input image,* and specify all the details that you want to preserve about the image (see examples below). As a bare minimum, Titan expects the prompt to describe the main object(s) of interest in the scene/image.

The level of details depends on your use case. In the images below you can see how richer prompts can help preserve more and more the details of the original image:

Original Image	Prompt	Output	Observation
	"a green iguana"		Good Prompt
	""		Bad Prompt: the model is not able to estimate the type of animal shown → we need the text prompt to describe the object/animal
	"a purple iguana"		Bad Prompt: The generated variation contains the animal in the prompt, but the prompt itself is not following the guidelines (i.e., a description of the INPUT image). Luckily, the model disregards the attribute "purple"
	"a green crocodile"		Bad Prompt: The generated variation contains the animal in the prompt, but the prompt itself is not following the guidelines (i.e., a description of the INPUT image).
	"two woman sitting on a park bench"		Good Prompt

	""		Not ideal Prompt, but the model is able to understand the scene and still create a meaningful output
	"two woman sitting on a park bench, one woman is wearing yellow trousers"		Good Prompt The model is able to preserve the yellow trousers, as specified in the prompt
	"food"		Ok Prompt, but the dish lost some of the features from the original image (i.e., what's in it)
	"a dish with some bread slices and arugula"		Good Prompt, the food inside the new dish resembles the same ingredients of the original image

Conclusion

Following the best practices for writing good prompts is key to generating high-quality images with Amazon Titan Image Generator models. Carefully crafting prompts by including vivid details, style keywords, negative prompts, and iterative refinement allows users to steer the image generation process.

In summary, when using Titan for image generation:

- Rely on descriptive wording, specific details, model parameters, and double quotes for text.

- For inpainting and outpainting, provide precise text about objects, colors, backgrounds, and context.

- Outpainting especially benefits from comprehensive scene descriptions.

- With image variations, comprehensively describe the input image to preserve key details.

Following these prompt engineering best practices, combine with your experience, will enable generating images that match desired styles, compositions, and emotions.

Thoughtful prompt design unlocks the full potential of Amazon Titan Image Generator models.