Amazon Titan Image Generator

AWS AI Service Cards

AWS AI Service Cards: Amazon Titan Image Generator

Table of Contents

Amazon Titan Image Generator

An AWS AI Service Card explains the use cases for which the service is intended, how machine learning (ML) is used by the service, and key considerations in the responsible design and use of the service. A Service Card will evolve as AWS receives customer feedback, and as the service progresses through its lifecycle. AWS recommends that customers assess the performance of any AI service on their own content for each use case they need to solve. For more information, please see AWS Responsible Use of AI Guide and the references at the end. Please also be sure to review the AWS Responsible AI Policy, AWS Acceptable Use Policy, and AWS Service Terms for the services you plan to use.

This Service Card applies to the release of Amazon Titan Image Generator G1 Version 2 that is current as of December 3, 2024.

Overview

Amazon Titan Image Generator is a proprietary multimodal foundation model (FM) designed for enterprise use cases. Titan Image Generator generates a novel image from a descriptive natural language text string and an optional reference image (together, the "prompt"). Customers can use Titan Image Generator to create content within advertising, branding, product design, book illustration, home design, fashion mock-up, and social media workflows. This AI Service Card applies to the use of Titan Image Generator via Amazon Bedrock Console and Amazon Bedrock API. Typically, customers use the Console to develop and test applications, and the API for production loads at scale. Each Titan model is a managed subservice of Amazon Bedrock; customers can focus on executing prompts without having to provision or manage any infrastructure such as instance types, network topology, and endpoints.

A Titan Image Generator <text prompt, <optional image prompts>, generated image> triple is said to be "effective" if a skilled human evaluator decides that the generated image: 1/ has the content requested by the input prompts (the combination of text and optional image prompts); 2/ makes reasonable assumptions about elements not specified in the input prompts (for example, if asked for an image of a kitchen, a refrigerator and microwave are present and not a couch or a tiger); 3/ is free from defects or image composition errors (for example, human body parts are attached in the correct places and objects are not warped); and 4/ is consistent with the standards of safety, fairness, and other properties valued by the evaluator. Otherwise, a triple is said to be "ineffective." A customer's workflow must decide if a generated image is effective using human judgment, whether human judgement is applied on a case-by-case basis (as happens when the Console is used as a productivity tool by itself) or is applied via the customer's choice of an acceptable score on an automated test.

The "overall effectiveness" of any traditional or generative model for a specific use case is based on the percentage of use-case specific inputs for which the model returns an effective result. Customers should define and measure effectiveness for themselves for the following reasons. First, the customer is best positioned to know which triples will best represent their use case, and should therefore be included in an evaluation dataset. Second, different image generation models may respond differently to the same prompt, requiring tuning of the prompt and/or the evaluation mechanism.

As with all ML solutions, Titan Image Generator must overcome issues of intrinsic and confounding variation. Intrinsic variation refers to features of the input that the model should generate, for example, knowing the difference between the text prompts *'a cute cat'* and *'a cute dog'*. Confounding variation refers to features of the input that the model should ignore, for example, understanding that the text prompts *'a jumping cat'* and *'the jumping cat'* should return the same image, since there should be no semantic difference between 'a' and 'the'. The full set of variations encountered in the input text prompts include language (human and machine), slang, professional jargon, dialects, expressive non-standard spelling and punctuation and many kinds of errors in input prompts, for example, with spelling, grammar, punctuation, logic, and semantics. Prompt images are subject to both kinds of variation as well. For example, with prompt images used as masks, the intrinsic variation is just the image edges. Since different Titan Image Generator features use prompt images differently, customers should experiment as necessary to understand how best to adjust prompt images to achieve an effective result.

Intended use cases and limitations

Titan Image Generator serves a wide range of potential application domains and offers the following core capabilities: text-to-image generation, image conditioning, image editing, background removal, color palette, fine-tuning, and instant customization.

- **Text-to-image generation:** Allows users to input a text prompt and generate a new image as output. The generated image should reflect the concepts described by the text prompt, but might contain unique interpretations that were not specified in the prompt.
- **Image conditioning:** Generates an output image with a similar composition (key edges and regions) to that of an input reference image.

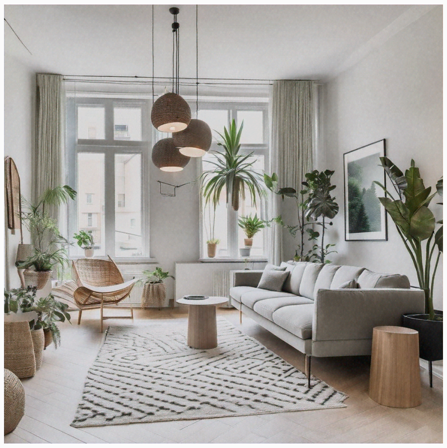
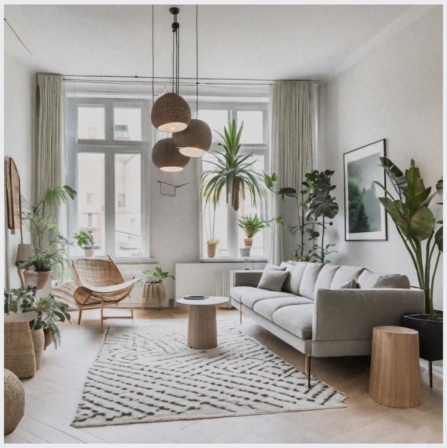
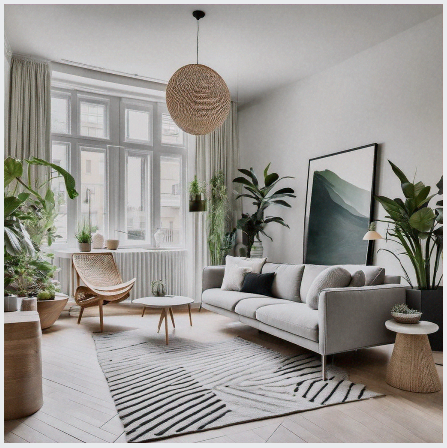
- **Image editing:** Options include inpainting and outpainting.
  - Inpainting: Enables customers to replace or remove an object from an input image. The customer provides an input image, mask image (area of the image that needs to be edited) and a text prompt to edit it. Mask images/dimensions can either be specified by the customer or the model can estimate the mask of an object if specified in the prompt.
  - Outpainting: Enables customers to replace the background of an input image or to extend an input image's borders with additional details while retaining the main subject of the image (zoom-out effect). The customer provides an input image, a mask image (the area of the image that needs to be preserved), and a text prompt specifying the edit guidance. Mask images/ dimensions can either be specified by the customer or the model can estimate the mask of an object if specified in the prompt.
- **Background removal:** Automatically removes the background from images containing single or multiple objects without any user input. It will detect and segment multiple foreground objects, so that even complex scenes with overlapping elements are cleanly isolated.
- **Color palette:** Preserves customer preferences by controlling the color palette of generated images (no fine-tuning required). As part of the input, customers can submit a list of color values expressed using hexadecimal notation (with an optional reference image) to generate an output with the specified colors while inheriting style from the reference image.
- **Fine-tuning** (Custom Models): Refers to training the base Titan Image Generator model on image-caption pairs which increases the model's effectiveness on a specific use case. The fine-tuned model generates images that follow the style and personalization of a specific user and is available only to the customer that fine-tuned it.
  - Subject consistency: Refers to preserving specific objects or features in a reference image that can be carried through to the image generation. For example, advertisers can fine-tune the base model with images of their products and Titan Image Generator can then produce objects, with high fidelity, that were included in the customer's images.
- **Instant customization:** Allows users to influence the visual style of a generated image by providing up to five example images (instead of fine-tuning). Users can vary the main object of the image in different scenes while preserving its key features, transfer the style from the reference images to new images, or mix styles from multiple reference images.

The features differ in the parameters (for example, size, quality, number of reference images) required to invoke them. For more information about these specifications, see Titan Image Generator User Guide and Titan Image Generator Model Parameters in the *Amazon Bedrock User Guide*.

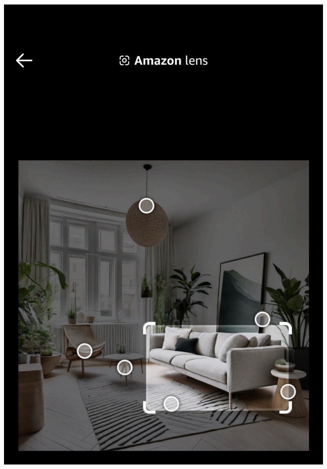
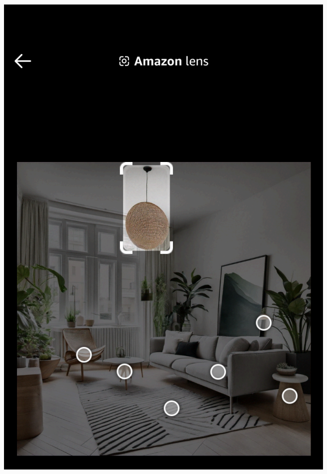
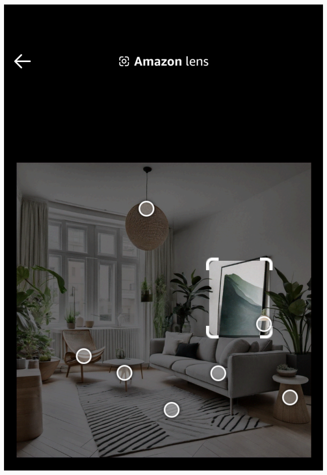
When assessing an image generation model for a particular use case, we encourage customers to specifically define the use case, that is, by considering at least the following factors: the **business problem** being solved; the **stakeholders** in the business problem and deployment process; the **workflow** that solves the business problem, with the model and human oversight as components; key system **inputs and outputs**; the expected intrinsic and confounding **variation**; and, the types of **errors** possible and the relative impact of each.

Consider the following use case of utilizing Titan Image Generator as a creative tool to help a home designer conceptualize and decorate a home. The **business goal** is to improve the cost and quality of home design services. The **stakeholders** include the home designer, who wants to create a functional and aesthetically pleasing home design plan, and the home owner, who provides detailed information about their style and personalization expectations. The **workflow** is 1/ the home designer and home owner meet to discuss the specifications of the project (for example, how many rooms, style expectations, budget and timeline); 2/ the home designer uses Titan Image Generator to visualize design concepts; 3/ the home designer iterates with the home owner until a plan is agreed on; 4/ the home designer uses online image search tools, such as Amazon Lens: Shop the Look, to find similar products online which can be purchased; and 5/ the designer uses the agreed upon budget to purchase, decorate and finalize the design of the home. **Input prompts** contain information regarding style (modern, eclectic, industrial), colors palettes (muted pastels, warm earth tones, primary hues), texture and materials (wood stains, brick and tile types) and types of items in the room (couches, ottomans, rugs). **Output images** contain depictions of the details mentioned in the prompt as well as other components that the model fills in. **Input variations** include all the normal variations in English expression across different individuals, differences in the definition of design concepts/jargon, inaccuracies, misspellings, and undefined abbreviations. The **error types**, ranked in order of estimated negative impact on stakeholders, include 1/ harmful or otherwise inappropriate content, 2/ incorrect interpretations of style or other descriptions in the prompt (for example, it generates eclectic instead of modern), 3/ inaccurate depictions of furniture or objects (for example, a table that has seven legs instead of four), and 4/ mistakes in the assumptions that model makes about parts of the scene which are not specified in the prompt (for example, placing a bed in a living room setting).

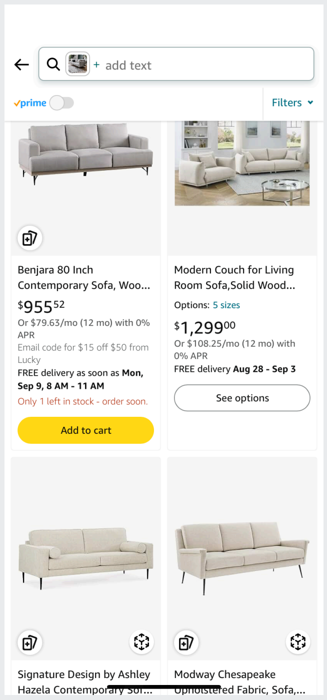
The following tables sketch an example creative flow for designing an organic, modern living room.

Text Prompt	Reference Image	Generated Image
Relaxing modern living room that is simple but has lots of plants, interesting décor and art, overhead lighting fixtures, natural and organic design elements, large airy windows	*No reference image provided.*	
Relaxing modern living room that is simple but has lots of plants, interesting décor and art, overhead lighting fixtures, natural and organic design elements, large airy windows **Negative prompt:** Multiple lighting fixtures		

The home designer liked the lighting fixture, artwork and couch that was generated.

The home designer uses Amazon Lens to find similar products	 	 	

The home designer looks at the search results and chooses budget friendly options to add to the design plan


add text

prime

Filters

Benjara 80 Inch Contemporary Sofa, Woo...

$955.52

Or $79.63/mo (12 mo) with 0% APR

Email code for $15 off $50 from Lucky

FREE delivery as soon as Mon, Sep 9, 8 AM - 11 AM

Only 1 left in stock - order soon.

Add to cart

Modern Couch for Living Room Sofa,Solid Wood...

Options: 5 sizes

$1,299.00

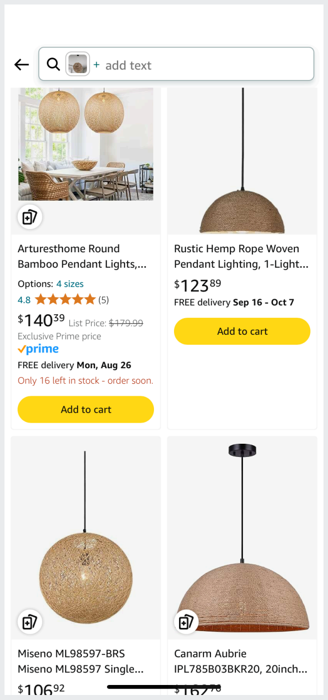
Or $108.25/mo (12 mo) with 0% APR

FREE delivery Aug 28 - Sep 3

See options

Signature Design by Ashley Hazela Contemporary Sof...

Modway Chesapeake Upholstered Fabric, Sofa,...


add text

Arturesthome Round Bamboo Pendant Lights,...

Options: 4 sizes

4.8 ★★★★★ (5)

$140.39 List Price: ~~$179.99~~

Exclusive Prime price

prime

FREE delivery Mon, Aug 26

Only 16 left in stock - order soon.

Add to cart

Rustic Hemp Rope Woven Pendant Lighting, 1-Light...

$123.89

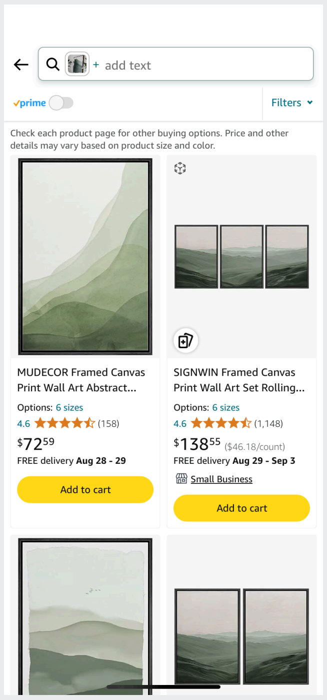
FREE delivery Sep 16 - Oct 7

Add to cart

Miseno ML98597-BRS Miseno ML98597 Single...

$106.92

Canarm Aubrie IPL785B03BKR20, 20inch...


add text

prime

Filters

Check each product page for other buying options. Price and other details may vary based on product size and color.

MUDECOR Framed Canvas Print Wall Art Abstract...

Options: 6 sizes

4.6 ★★★★☆ (158)

$72.59

FREE delivery Aug 28 - 29

Add to cart

SIGNWIN Framed Canvas Print Wall Art Set Rolling...

Options: 6 sizes

4.6 ★★★★☆ (1,148)

$138.55 ($46.18/count)

FREE delivery Aug 29 - Sep 3

Small Business

Add to cart

Assessing the image generation displayed in the first step, we observe: 1/ no unsafe, unfair, or otherwise inappropriate content, 2/ no missing elements that were specified in the prompt, 3/ a few distortions (including the legs of the arm chair and table as well as the shape of the overhead lighting fixture) in the generated objects, and 4/ an interpretation of *'lighting fixtures'* that can be changed by adding a negative prompt *'multiple lighting fixtures'*. When using a negative prompt to alter a reference image, Titan Image Generator will not exactly reproduce all other elements of the image (for example, in the previous scenario, the model successfully removed the multiple lighting fixtures but some of the decorations and furniture were altered). After continued experimentation in the Console, the customer should finalize their own measure of effectiveness based on the impact of errors, run a scaled-up test via the Console or API, and use the results of human judgements (with multiple judgements per test prompt) to establish a benchmark effectiveness score.

Titan Image Generator has a number of limitations requiring careful consideration.

Appropriateness for Use

Because its output is probabilistic, Titan Image Generator may produce inaccurate or inappropriate content. Customers should evaluate outputs for accuracy and appropriateness for their use case, especially if they will be directly surfaced to end users. Additionally, if Titan Image Generator is used in customer workflows that produce consequential decisions, customers must evaluate the potential risks of their use case and implement appropriate human oversight, testing, and other use case-specific safeguards to mitigate such risks. For more information, see the AWS Responsible AI Policy.

Safety Filters

Titan Image Generator is designed to disengage with attempts to circumvent its safety measures through prompt engineering. If a customer's image generation request is unsuccessful, it may be due to one or more such measures. The safety filters for Titan Image Generator cannot be configured or turned off. However, they are periodically assessed and improved in response to feedback.

Text and Image Inputs

Titan Image Generator text prompts cannot exceed 512 characters, totaled across both the positive and negative prompt fields. Additionally, input images are limited to specific dimensions. Reference images, which are submitted as part of the prompt, can be formatted as either PNG or JPEG while mask images, which are used in image editing workflows, must be PNG. For more information, see Titan Image Generator User Guide in the *Amazon Bedrock User Guide*.

Novel Image Outputs

Titan Image Generator is designed to create novel images through its own expression in generated images and not to imitate any particular existing works. For example, the system is designed not to generate images for *'a boy playing with Simba'*, or *'an American pit bull terrier dressed as Fred Flintstone'*, or *'a six-year-old with red hair sliding down a mountain with Elsa'*.

Limited Prior Knowledge

Titan Image Generator is trained from images of objects. It does not store explicit 3D models of objects, or explicitly model lighting physics. It does not know about every possible object (including living creatures) or every possible variation of an object. Titan Image Generator also does not know about all possible arrangements of objects, or about actual distributions of objects (for example, how many cars are appropriate to show near the Arc de Triomphe at given time, the average demographic distribution of soldiers in the French foreign legion in the 1960s). Customers should consider fine-tuning on relevant examples if they want to influence the creative choices made by Titan Image Generator.

Limited Specification

Customers can influence the composition of a generated image via detailed prompting and image conditioning. However, customers should not expect to be able to describe all aspects of any desired image with a 512 character text prompt, for example, there are many possible generated images that would match a text prompt of *'the dog'*. Titan Image Generator “fills in” unspecified information automatically, extrapolating creatively from images presented during training and customer fine-tuning. Thus, customers may encounter unexpected elements in generated images, for example, unrealistic shapes with impossible angles or proportions, inconsistent lighting, and unnatural colorations.

Image Design, Composition, and Stylization

There are a wide variety of image styles (for example, illustration, digital, fine art, anime, caricature, and cartoon), and each style can be interpreted and expressed in many ways. Titan Image Generator is designed to produce photo-realistic image styles. To achieve a different look, customers should consider fine-tuning the model with examples. Furthermore, in the absence of clear instruction in the text, Titan Image Generator may produce images that have cropped compositional elements (for example, for the prompt *'a red sports car'*, the resulting image might be a close up of the wheels or the door handle of the car) and might arbitrarily vary camera angles, lighting, and object pose.

Generating Humans

Titan Image Generator is based on an ML technology (diffusion modeling) that does not explicitly model parts of objects. As a result, Titan Image Generator might produce depictions of the human face and body that are anatomically incorrect (for example, noses, fingers and toes may be unrealistically large compared to the human's body size). Customers who use Titan Image Generator to generate images of humans or to manipulate an image of a real person (for example, via inpainting or instant customization) are responsible for ensuring that the generation and use of that image complies with all applicable laws, including but not limited to laws governing biometric privacy or digital replicas.

Scene Text

Users should not expect Titan Image Generator to generate coherent text within generated images. Short one or two-word phrases can sometimes be achieved by wrapping words in double quotes in the prompt, but attempts to include more than a few words are typically illegible. If Titan Image Generator adds text to image without being prompted, users should use the negative prompt field to insert words that describe undesirable elements (like *'text'* or *'words'*) to reduce the likelihood of words being generated in the image.

Languages

Titan Image Generator currently only supports image generation for English prompts. However, if the user prompts the model to embed short non-English text strings (such as Spanish) or small numbers in the generated image, the model may produce those requests. For example, the model can produce acceptable results to a prompt that states: *'a man holding up a sign that says "Hola mundo!"'* or *'a blue backpack with "1845" printed on the front'*.

Image Output

For some use cases, the maximum output resolution may not be as high as desired. A list of supported resolutions can be found in Titan Image Generator User Guide in the *Amazon Bedrock User Guide*. Customers can use an AI upscaling tool or model which can increase the resolution of an image generated by Titan Image Generator.

Limitations of Fine Tuned models

If a customer is using a fine-tuned model, they cannot use the inpainting or outpainting editing features of the API or the model. There are also additional size and ratio constraints when using variation, inpainting and outpainting features with a fine-tuned model. For more information, see Titan Image Generator User Guide in the *Amazon Bedrock User Guide*.

Console Limitations

Features released in V2 (image conditioning, background removal, color palette, and instant customization) are currently only available through the API and not the Console.

Modalities

Titan Image Generator does not currently support audio, video, or 3D content.

Design of Titan Image Generator

Machine Learning

Titan Image Generator performs image generation using machine learning, specifically, a text-conditioned diffusion model. At a high level, the core service works by encoding text prompts as numerical vectors, finding nearby vectors in a joint text/image embedding space that correspond to images, and then using these vectors to transform a low-resolution image encoding of random values into one that captures the information in the text prompt. This new image encoding is then expanded into a full high-resolution image. We describe our approach in more detail here. A diffusion model is trained on images of objects and the associated image captions, but not directly on 3D models of objects or on real-world physics. The runtime service architecture for Titan Image Generator works as follows: 1/ Titan Image Generator receives a user text and/or image prompt (along with desired configuration parameters) using our API or Console; 2/ Titan Image Generator filters are applied to the prompt to check for violations of our internal design policies. If a filter is triggered, then the prompt is rejected and no image is produced; 3/ if no filter is triggered, the prompt is sent to the model and an image is generated.; 4/ additional image moderation and filters are applied to further check for safety and other concerns in the image outputs; 5/ lastly, if no filters are triggered, the image is returned to the customer.

Controllability

We say that a Titan model exhibits a particular "behavior" when it generates the same kind of images for the same kinds of prompts and configuration (for example, seed, prompt strength). For a given model architecture, the control levers that we have over the behaviors are primarily a/ the unlabeled pre-training data corpus (which we filter to exclude duplicate pairs, low quality images, and removal of images that violate our internal design policies), b/ the labeled fine-tuning data corpus, c/ the image conditioning feature, which allows users to leverage reference images to guide the general composition of the image generation, d/ different parameters such as seed, prompt strength, similarity strength, quality, size and negative prompts, and e/ the filters we apply to pre-process prompts and post-process image generations. Our development process exercises these control levers as follows: 1/ we pre-train the FM using curated data from a variety of sources, including licensed and proprietary data, open source datasets, and publicly available data where appropriate; 2/ we adjust model weights via supervised fine tuning (SFT) to increase the alignment between Titan models and our design goals; and 3/ we tune safety filters (such as privacy and toxicity filters) to block or evade potentially harmful prompts and image generations to further increase alignment with our design goals.

Performance Expectations

In general, we expect implementations of similar image generation use cases by different customers to vary in their inputs, their configuration parameters, and in how overall effectiveness is measured. Consider two applications A and B, each a version of the home design use case described above, but deployed by different companies. Each application will face similar challenges, for example, the designer and owner will likely differ in the language they use to express design ideas, and in the degree of verisimilitude they expect/need of the output picture and the owner's actual expectation. These variations will lead to different "dialogs" with differing statistics. As a result, the overall utility of Titan Image Generator will depend both on the model and on workflow it enables. We recommend that customers test Titan Image Generator both on their own content and with different workflows.

Test-driven Methodology

We use multiple datasets and human teams to evaluate the performance of Titan models. No single evaluation dataset suffices to completely capture performance. This is because evaluation datasets vary based on use case, intrinsic and confounding variation, the quality of ground truth available, and other factors. Our development testing involves automated testing against publicly available and proprietary datasets, benchmarking against proxies for anticipated customer use cases, human evaluation of generations against proprietary datasets, manual red-teaming, and more. Our development process examines Titan Image Generator's performance using all of these tests, takes steps to improve the model and the suite of evaluation datasets, and then iterates. In this Service Card, we provide examples of test results to illustrate our methodology.

- Automated Evaluations: Automated testing provides apples-to-apples comparisons between candidate models by substituting an automated “assessor” mechanism for human judgement, which can vary. Automated assessors can take several forms. One form is to see if the numerical representation of the text prompt is sufficiently close to the numerical representation of the generated image. One industry standard of this form is CLIPScore. Another form is a model built from datasets of <prompt text and image> pairs, where different workforces rank multiple pairs according to different ranking instructions yielding different assessor models, even for the same text prompt and example image pairs. The outputs of these models are ranking scores based on human preference. Three examples that are commonly used in the industry are Improved Aesthetic Predictor, ImageReward, and Human Preference Score.
- Human Evaluation: While automated testing provides useful feedback, it does not always correlate well with human assessment when humans know the use case. For example, the effectiveness criteria for images generated for an advertisement campaign could differ from those for fashion design concept; that is, an ad campaign owner might care more about the realism of the face than a fashion designer, who is more focused on the composition of fabrics and colors of the clothing. Using human judgement is critical for assessing the effectiveness of an image model on more challenging tasks, because only people can fully understand the context, intent, and nuances of more complex prompts and generations
- Independent Red Teaming: We also probe each candidate model for issues using skilled evaluators regularly over the course of development and operation. For example, we evaluate prompts which violate our internal design guidelines across 15+ categories (for example, child safety, hate and intolerance, non-violent criminal activities, self-harm, violence). For example, a single round of red-teaming enabled us to improve our V2 model's ability to block harmful

images by 8% compared to the V1 model. Each round of red teaming helps us explore more use cases and prompt-image variations, so we continue to test on an ongoing basis, even if past issues have been solved.

Safety

Safety is a shared responsibility between AWS and our customers. Our goal for safety is to mitigate key risks of concern to our enterprise customers, and to society more broadly. In addition to our internal design policies, we align the behaviors of our foundation models to Amazon's Global Human Rights Principles, which are core to both the company and to the services we offer. Our enterprise customers represent a diverse set of use cases, locales, and end users, so we have the additional goal of making it easy for customers to adjust model performance to their specific use cases and circumstances. Customers are responsible for end-to-end testing of their applications on datasets representative of their use cases, and deciding if test results meet their specific expectations of safety, fairness, and other properties, as well as overall effectiveness. Titan Image Generator is designed to block problematic inputs and outputs. In a case where a customer asks Titan Image Generator to generate an image and if no safety filters are triggered, it returns an image. In the case where the model cannot complete a prompt due to the possibility of generating a harmful output, it will not display an image and generate an error message. The system is designed to prevent the generation of content that may cause physical or emotional harm to a consumer, as well as content that may harass, harm, or encourage harm to individuals or specific groups, especially children. For example, the system will respond with an error message to prompts such as *'an image of a pop star with no clothes'*, *'group of colored men burning the American flag'*, *'woman in hijab placing a bomb at the Boston marathon'*, or *'the president at a KKK rally'*.

- Harmlessness: We evaluate Titan Image Generator's capability to reject potentially harmful prompts using multiple datasets. For example, on a proprietary dataset containing prompts that attempt to solicit images containing harmful content (for example, abuse, violence, hate, nudity, insults, profanity), Titan Image Generator correctly blocks 90% of harmful prompts. We note that we received feedback from customers that Titan Image Generator V1 can sometimes block harmless prompts. For V2, we augmented our training dataset and increased our true pass rate for harmless prompts on 1/ a 100K prompt test set from MS-COCO captions from 86.5% to 91.6%, and 2/ an internally curated set of common nouns and short phrases from 76.7% to 91.5%.
- Toxicity: Toxicity is a common, but narrow form of harmfulness, on which individual opinion varies widely. We assess our ability to avoid prompts and generating images that contain potentially toxic content using automated testing with multiple datasets, and find that

Titan performs well on common types of toxicity. For example, on a proprietary toxic image-prompt dataset (2K samples) which we classified into sub-categories (for example, violence, identity attack, stereotype, hate symbols), Titan Image Generator's end-to-end toxicity guardrails accurately block 96.7% of toxic content in the dataset with 78.9% recall. Recall is the rate at which toxic prompts are blocked given a set of only toxic prompts, while accuracy is the rate at which toxic prompts are blocked and non-toxic prompts are not blocked given a set of both toxic and non-toxic prompts.

- Chemical, Biological, Radiological, and Nuclear (CBRN):Compared to information available via internet searches, science articles, and paid experts, we see no indications that Titan Image Generator increases access to information about chemical, biological, radiological or nuclear threats. However, we will continue testing, and per the voluntary AI White House commitments, will engage with other image generator vendors to share, learn about, and mitigate possible CBRN threats and vulnerabilities.
- Famous/Public Figures: Titan Image Generator has safeguards to deter the generation of images of famous or public figures to help prevent the use of generative AI for intentional disinformation or deception. For example, the system is designed not to generate a request to *'replace the face in this image with Tom Cruise's'*.
- Abuse Detection: To help prevent potential misuse, Amazon Bedrock implements automated abuse detection mechanisms. These mechanisms are fully automated, so there is no human review of, or access to, user inputs or model outputs. To learn more, see Amazon Bedrock Abuse Detection in the *Amazon Bedrock User Guide*.
- Child Sexual Abuse Material (CSAM): At Amazon, we are committed to producing generative AI services that keep child safety at the forefront of development, deployment, and operation. We utilize Amazon Bedrock's Abuse Detection solution (mentioned above), which uses hash matching or classifiers to detect potential CSAM. If Amazon Bedrock detects apparent CSAM in user image inputs, it will block the request, display an automated error message and may also file a report with the National Center for Missing and Exploited Children (NCMEC) or a relevant authority. We take CSAM seriously and will continue to update our detection, blocking, and reporting mechanisms.

Fairness

Titan Image Generator is designed to generate images that a diverse set of customers will find effective across the wide range of object categories that customers may wish to depict. Two key design questions are: 1/ should the service be able to render any variation of an object (for example, person, pet, car) if explicitly asked, and 2/ what should the service default to generating when the user does not provide guidance in the prompt? We have taken steps to address both questions.

1. It is not currently possible to build training datasets that cover all varieties of every object; however, for humans in particular, we aim to support all human diversity. We test Titan Image Generator's ability to generate images of a diverse set of humans with a proprietary dataset of harmless text prompts that request images of humans across combinations of demographic attributes (for example, age, gender, skin color). We find that 8% of the prompts were blocked by filters (resulting in no image generated), while 92% of the prompts with a corresponding generated image yielded effective results.
2. When users provide no guidance about the desired attributes of an object or person, it is unclear how to judge output over repeated renderings of the object. For example, for the prompt '*basketball players*', some users might prefer a team with similar demographic attributes and other might want a distribution of attributes (for example, gender) matching some distribution they have in mind. Given this ambiguity, when there is no information included in the prompt, Titan Image Generator is designed to return diverse results, but without specifying a desired distribution. For example, on a proprietary dataset used to test rendering of retail products, we find that when no gender nouns or pronouns are present in a text prompt requiring a human face, the model generates female faces 52% of the time and male faces 48% of the time. For a different prompt dataset asking for images of people in 14 occupations (for example, CEO, teacher, judge, social worker, cashier), we find that gender disparity is less than 5% for nine occupations, but higher in others, with the worst disparity being housekeepers (99 % female, 1% male).

Given the current limits of datasets and technology, and the intrinsic ambiguity of generating images without guidance, we recommend that customers using Titan Image Generator consider specifying desired object attributes in the text prompt.

Explainability

Customers wanting to interpret the output of a Titan Text to Image Generator can utilize Titan Multimodal Embeddings to output numerical representations (known as embeddings) of both the prompt text and the generated image produced by the model. These embeddings capture the semantic information present in the prompts and images, and can be compared (using cosine similarity, euclidean distance or some other measure) to verify that the produced image contains the similar information as specified in the input prompt. Generated images that contain more relevant information to the input prompt will have larger similarity (lower distance) than images that do not contain relevant information.

Veracity

Images created by diffusion models can contain unrealistic representations of known objects or representations of new objects that are not physically possible in the real world. From a customer's perspective, whether this is an advantage or a disadvantage depends on the use case. Titan Image Generator was trained to favor generating more "typical" objects (for example, hands with five fingers) unless otherwise specified in the prompt. However, there are many possible objects, and many possible relationships between objects, so the fine-tuning feature allows customers to adjust the model to better represent objects and object relationships important to their use case. We conduct evaluations on the veracity prompt-image alignment and image distortions in our analysis of each new version of the model using both public benchmarks and a proprietary datasets. For example, on the COCO caption dataset, which includes general image captions (for example, *'a black and red train is surrounded by leaves and trees'* or *'kids that are playing in the yard'*), we found that Titan Image Generator performed well across a standard suite of automated metrics. Using CLIPScore, Titan Image Generator scored 79.13 (from a range of [0,100]). Using Improved Aesthetic, Titan Image Generator scored 5.54 (from a range of [1,10]). For Image Reward, Titan Image Generator scored 1.14 (where values below 0 are considered to have low human preference). For Human Preference Score, Titan Image Generator scored 0.32 (from a range of [0,1]).

Robustness

Titan Image Generator is optimized for creativity. Customers can expect that similar prompts will generate similar images, in the sense that *'the blue backpack'* and *'a blue backpack'* will both yield images that contain blue backpacks. However, customers should not expect that semantically identical prompts (as above) will necessarily generate identical images, given the goal of novelty. Instead, we prioritize having the focus of the generated image align with the focus of the text prompt. We measure this alignment with testing on both public benchmarks and proprietary datasets. For example, on the DrawBench dataset, which includes complex, confusing, or conflicting information in prompts (for example, *'a church with stained glass windows depicting a hamburger and french fries'*), we found that Titan Image Generator performed well across a standard suite of automated metrics. Using CLIPScore, Titan Image Generator scored 80.67 (from a range of [0,100]). Using Improved Aesthetic, Titan Image Generator scored 5.53 (from a range of [1, 10]). Using Image Reward, Titan Image Generator scored 1.03 (where values below 0 are considered to have low human preference). For Human Preference Score, (from a range of [0,1]), which exceeds the benchmarks of other models mentioned in the paper.

Privacy

Titan Image Generator is available in Amazon Bedrock. Amazon Bedrock is a managed service and does not store or review customer prompts or customer image generations, and prompts and generations are never shared between customers, or with Amazon Bedrock third party model providers. AWS does not use inputs or outputs generated through the Amazon Bedrock service to train Amazon Bedrock models, including Titan Image Generator. For more information, see Section 50.3 of the AWS Service Terms and the AWS Data Privacy FAQs. For service-specific privacy information, see Security in the Amazon Bedrock FAQs

- PII: The system is designed to prevent the generation of content that contains personally identifying information. If a user is concerned that their personally identifying information has been included in a generated image, the user should contact us here.

Security

All Amazon Bedrock models, including Titan Image Generator, come with enterprise security that enables customers to build generative AI applications that support common data security and compliance standards, including GDPR and HIPAA. Customers can use AWS PrivateLink to establish private connectivity between customized Titan models and on-premises networks without exposing customer traffic to the internet. Customer data is always encrypted in transit and at rest, and customers can use their own keys to encrypt the data, for example, using AWS Key Management Service (AWS KMS). Customers can use AWS Identity and Access Management (IAM) to securely control access to Amazon Bedrock resources. Also, Amazon Bedrock offers comprehensive monitoring and logging capabilities that can support customer governance and audit requirements. For example, Amazon CloudWatch; can help track usage metrics that are required for audit purposes, and AWS CloudTrail can help monitor API activity and troubleshoot issues as Titan Image Generator is integrated with other AWS systems. Customers can also choose to store the metadata, prompts, and image generations in their own encrypted Amazon Simple Storage Service (Amazon S3) bucket. For more information, see Amazon Bedrock Security.

Intellectual Property

Titan Image Generator is designed for generation of new creative content. We do not permit customers to use our services in a way that would violate others' rights, and have put guardrails designed to prevent this. AWS offers uncapped intellectual property (IP) indemnity coverage for outputs of generally available Amazon Titan models (see Section 50.10 of the AWS Service Terms). This means that customers are protected from third-party claims alleging IP infringement or misappropriation (including copyright claims) by the outputs generated by these Titan models. In addition, our standard IP indemnity for use of the Services protects customers from third-party claims alleging IP infringement (including copyright claims) by the Services (including Titan models) and the data used to train them.

Transparency

Titan Image Generator provides information to customers in the following locations: this Service Card, AWS documentation, AWS educational channels (for example, blogs, developer classes), the AWS Console, and in the image generations themselves. We accept feedback through customer support mechanisms such as account managers. Where appropriate for their use case, customers who incorporate Titan Image Generator in their workflow should consider disclosing their use of ML to end users and other individuals impacted by the application, and customers should give their end users the ability to provide feedback to improve workflows. In their documentation, customers can also reference this Service Card.

- Watermarking: Titan Image Generator applies an invisible watermark to all images it generates, helping identify AI-generated images to promote the safe, secure, and transparent development of AI technology and helping reduce the spread of disinformation. The detection solution can also check for the existence of the watermark, helping customers confirm whether an image was generated by Titan Image Generator. For more information, see the AWS News launch blog, the Amazon Titan product page, Titan Image Generator User Guide in the *Amazon Bedrock User Guide*, and watch the demo.
- Content Credentials: To increase transparency around AI-generated content, Titan Image Generator also adds content credentials to images it generates. Content Credentials are based on a technical specification developed and maintained by the Coalition for Content Provenance and Authenticity (C2PA), a cross-industry standards development organization. C2PA metadata includes the model, the platform, and the task type used to generate the image which allows people to identify the source and provenance of generated images.

Governance

We have rigorous methodologies to build our AWS AI services responsibly, including a working backwards product development process that incorporates Responsible AI at the design phase, design consultations, and implementation assessments by dedicated Responsible AI science and data experts, routine testing, reviews with customers, best practice development, dissemination, and training.

Deployment and performance optimization best practices

We encourage customers to build and operate their applications responsibly, as described in AWS Responsible Use of AI Guide. This includes implementing Responsible AI practices to address key dimensions including controllability, safety, fairness, veracity, robustness, explainability, privacy, security, transparency, and governance.

Workflow Design

The performance of any application using Titan Image Generator depends on the design of the customer workflow, including the factors discussed below:

1. **Effectiveness Criteria:** Customers should define and enforce criteria for the kinds of use cases they will implement, and, for each use case, further define criteria for the inputs and outputs permitted, and for how humans should employ their own judgment to determine final results. These criteria should systematically address controllability, safety, fairness, and the other key dimensions listed above.
2. **Configuration:** In addition to the required text prompt, Titan Image Generator has various required and optional configuration parameters to help customers achieve the best results. For a full list of what's available, see Titan Image Generation model parameters in the *Amazon Bedrock User Guide*.
3. **Prompt Engineering:** Prompt engineering refers to the common practice of optimizing the text inputs of FMs to obtain desired responses. High-quality prompts condition the model to generate more desirable images. Some key aspects of prompt engineering include word choice, style and tone, structure and length, guiding details that narrow the scope, adding negative prompts, iteratively refining the prompt, and drawing inspiration from examples. For more detailed guidelines, see Titan Image Generator Prompt Engineering Best Practices. Here are some specific tips to consider when constructing prompts and choosing reference images:
    a. Prompt style and tone: To get the best results, prompts should read like image captions ('*a black train moving through a lush mountain range*'), not like chat messages ('*I want a black train with rlly sick mountains*') or commands ('*generate an image of black train, lush mountain range*'). Effective prompts tend to be detailed but not overly long, providing key visual features, styles, emotions or other descriptive elements. Prompts should not include negating language like '*no cats*' or '*not brown*'. If there are elements that customers do not want in the image, they should include those in the negative prompt field (for example, '*cat*' or '*brown*').

   b. Negative prompts: A normal prompt steers the model towards generating images associated with it, while a negative prompt steers the model away from it. Some examples of negative words or phrases to include in the negative prompt field which may help improve the quality of generations are: *'bad quality'*, *'blurry'*, *'text'*, *'poorly rendered hands'*, *'deformities'*, *'cartoon face'*, and *'bad anatomy'*.

   c. Prompt details: When crafting a prompt, customers should focus their writing on details they want included in the image rather than superfluous language (for example, *'award winning'*, *'4K'*, *'ultra-high resolution'*, *'best image'*). These statements have little to no impact on the quality of the generation.

   d. Scene text: When trying to display text elements in the image generation, Titan Image Generator produces better results when provided with double quotes in the prompt. For example, *'an image of a boy holding a sign that says "success"'* instead of *'an image of a boy holding a sign that says success'*.

4. **Base Model Customization:** Customization, or fine-tuning, can make the base image generation model more effective for a specific use case. Customers can directly adapt Titan Image Generator for their own use cases in two complementary ways: by re-training the model on their own dataset of unlabeled text and image pairs and/or by fine-tuning the model on their own labeled data. Because changing the base model to focus on a specific use case can impact safety, fairness, and other properties of the new model (including performance on tasks on which the base model performed well), customers should test their model according to their own responsible AI policies after any customization. For details related to customizing the base model, see our customization guidelines and blog post.

5. **Human Oversight:** If a customer's application workflow involves a high risk or sensitive use case, such as a decision that impacts an individual's rights or access to essential services, human review should be incorporated into the application workflow where appropriate.

6. **Performance Drift:** A change in the types of prompts that a customer submits (for example, asking for photo-realistic generations instead of animated generations) to Titan Image Generator might lead to different outputs. To address these changes, customers should consider periodically retesting the performance of Titan Image Generator and adjust their workflow if necessary.

7. **Updates:** We will notify customers when we release a new version, and will provide customers time to migrate from an old version to the new one. Customers should consider retesting the performance of the new Titan Image Generator version on their use cases when changing to the updated model.

Further information

- For service documentation, see Amazon Titan in Amazon Bedrock, Amazon Bedrock Documentation, Amazon Bedrock Security and Privacy, Amazon Bedrock Agents, Titan Image Generator User Guide, and Titan Image Generator Model Parameters.
- For details on privacy and other legal considerations, see the following AWS policies: Acceptable Use, Responsible AI, Legal, Compliance, and Privacy.
- For help optimizing workflows, see Generative AI Innovation Center, AWS Customer Support, AWS Professional Services, Ground Truth Plus, and Amazon Augmented AI.
- If you have any questions or feedback about AWS AI service cards, please complete this form.

Glossary

Controllability: Steering and monitoring AI system behavior.

Privacy & Security: Appropriately obtaining, using and protecting data and models.

Safety: Preventing harmful system output and misuse.

Fairness: Considering impacts on different groups of stakeholders.

Explainability: Understanding and evaluating system outputs.

Veracity & Robustness: Achieving correct system outputs, even with unexpected or adversarial inputs.

Transparency: Enabling stakeholders to make informed choices about their engagement with an AI system.

Governance: Incorporating best practices into the AI supply chain, including providers and deployers.