Generative AI Prompting for Image Models

A Complete 10-Chapter AI Course

Topics covered:

How diffusion models actually work, and why image prompting is descriptor composition, not instruction-giving

Midjourney, Stable Diffusion, and DALL-E — three tools, three trade-offs

Universal techniques, honest failure modes, and real ethical questions

Exercises: 30 hands-on scenarios with worked solutions

Format: A4 · Dark-theme code examples

Single standalone course · the image-generation sibling prompt1 never had

Philip Osztromok · Generated with Claude

Table of Contents

CHAPTER 1 OF 10

Why Image-Model Prompting Is a Different Skill

Generative AI Prompting for Image Models

Chapter 1 · Why Image-Model Prompting Is a Different Skill

`prompt1` already covers prompting in real depth — clarity, context, constraints, role prompting, few-shot examples, chain-of-thought, iterative refinement. If prompting an image model were just "the same skill applied to pictures," this course wouldn't need to exist; it could be a single bonus chapter tacked onto the end of `prompt1`. It isn't, because the thing being prompted is a genuinely different kind of system.

What prompt1 Actually Assumes

`prompt1-2`'s own anatomy of a good prompt — Clarity, Context, Constraints, Output Format — rests on one load-bearing assumption: that the model being prompted can *follow an instruction*. Claude reads "summarize this in three bullet points, using a formal tone" and does something recognizable as obeying that instruction, because Claude was trained, via RLHF and instruction-tuning, specifically to map instructions to compliant behavior. Every technique `prompt1` teaches — role prompting (`prompt1-3`), few-shot examples (`prompt1-4`), chain-of-thought (`prompt1-5`) — is a way of shaping *how* an instruction-following system carries out an instruction.

An image model was never trained to do that.

What an Image Model Was Actually Trained to Do

Tools like Midjourney, Stable Diffusion, and DALL-E are built on **diffusion models** (the full mechanism is `imgai1-2`'s own subject — this chapter only needs the shape of the idea). During training, the model sees millions of (image, text caption) pairs and learns a statistical association between patches of text and patches of visual pattern — "a photo of a golden retriever" pulls the generation process toward pixel arrangements statistically associated with that caption in the training data. There is no step anywhere in that process resembling "read this sentence, form an understanding of what is being asked, and comply." The model has no representation of your request as a *request* at all — only as a string of text to condition image generation on.

THIS IS NOT A MINOR IMPLEMENTATION DETAIL

It's the reason a prompt like "please don't include any text in the image, that would be great, thanks" performs no better — often worse, from the extra noise words — than just `no text` as a negative prompt (`imgai1-7`).

> Politeness, framing, and instructional phrasing are wasted effort on a system with no concept of being instructed. This isn't a stylistic quirk to work around; it follows directly from what the model was actually trained to do.

Descriptor Composition, Not Instruction-Giving

If an image prompt isn't an instruction, what is it? The working model this course builds toward, starting concretely in `imgai1-3`: a prompt is closer to a **dense list of descriptors** — subject, style, composition, lighting, medium — than to a sentence directed at a listener. "Write a poem about the ocean" (an instruction, aimed at prompt1's own kind of model) and "a weathered lighthouse at dusk, oil painting, dramatic lighting, wide shot" (a descriptor composition, aimed at this course's own kind of model) are doing fundamentally different jobs, even though both are grammatically similar-looking strings of English.

	PROMPT1 (CLAUDE / LLMS)	THIS COURSE (IMAGE MODELS)
Underlying mechanism	Instruction-tuned transformer, trained to comply with requests	Diffusion model, trained to associate text embeddings with visual patterns
What a prompt "is"	A request, directed at a system capable of understanding it	A dense set of descriptors conditioning a generation process
Politeness / framing	Can genuinely shift tone and compliance	Functionally inert — no comprehension to appeal to
Core skill	Clarity, context, constraints (prompt1-2)	Descriptor vocabulary, weighting, tool-specific syntax
Refinement style	Conversational back-and-forth (prompt1-6)	Tool-dependent — from parameter tweaking to genuine conversation (imgai1-4 through imgai1-6)

Three Tools, Three Real Trade-offs

This course doesn't teach one image model — it teaches three, deliberately chosen because each makes a genuinely different trade-off, not because "more tools" is inherently better coverage:

- **Midjourney** (`imgai1-4`) — cloud-only, Discord-based, a distinctive bracketed `--parameter` syntax, and a well-documented painterly/stylized bias baked into the model itself.
- **Stable Diffusion** (`imgai1-5`) — open-source, can run entirely locally, and exposes real technical knobs (CFG scale, sampler choice, checkpoints, LoRAs) no other tool in this course offers.
- **DALL-E** (`imgai1-6`) — integrated directly into ChatGPT's own conversational interface, which makes it, genuinely, the one tool in this course closest to `prompt1`'s own territory. It's covered last specifically so that convergence lands as a real observation, not a starting assumption.

> THIS COURSE'S OWN ROADMAP

`imgai1-2` makes the diffusion mechanism concrete. `imgai1-3` builds the shared descriptor vocabulary every later chapter assumes. `imgai1-4` – `imgai1-6` cover the three tools in turn. `imgai1-7` collects techniques that recur across tools despite differing syntax. `imgai1-8` explains failure modes mechanically rather than hand-wavily. `imgai1-9` is a real ethics chapter. `imgai1-10` is a worked capstone.

Hands-On Exercises

EXERCISE 1

Explain, using this chapter's own material on how a diffusion model is trained, why an image model has no representation of a prompt as a "request" the way an instruction-tuned LLM like Claude does.

View solution

EXERCISE 2

Using this chapter's own comparison table, explain why polite or instructional phrasing ("please don't include any text, that would be great") performs no better than a blunt negative-prompt-style phrase, while the same courtesy genuinely can help with a Claude prompt.

View solution

EXERCISE 3

This chapter deliberately covers DALL-E last, after Midjourney and Stable Diffusion. Explain, using the chapter's own reasoning, why that ordering was a deliberate choice rather than an arbitrary one.

View solution

CHAPTER 1 QUICK REFERENCE

- LLMs (prompt1) are instruction-tuned; image models are diffusion models trained to associate text with visual patterns — no comprehension of a request exists in either case, but the LLM case at least has instruction-compliance behavior trained in
- An image prompt is closer to **descriptor composition** (subject, style, composition, lighting, medium) than to instruction-giving
- Politeness/framing is functionally inert on a diffusion model — it has nothing to appeal to
- Three tools, three trade-offs: **Midjourney** (cloud, stylized bias) · **Stable Diffusion** (open-source, technical control) · **DALL-E** (conversational, closest to prompt1's own territory)

- **Next chapter:** How Diffusion Models Actually Work

CHAPTER 2 OF 10

How Diffusion Models Actually Work

Generative AI Prompting for Image Models

Chapter 2 · How Diffusion Models Actually Work

The History of AI trilogy's own closing chapter, `historyai3-8`, named "diffusion models" in passing as one of the developments bringing that course up to the present day, alongside multimodal AI and the AGI/alignment debate — a namecheck, not an explanation. This chapter delivers the explanation, and does it specifically because `imgai1-1`'s central claim — that an image prompt is descriptor composition, not instruction-giving — is only a slogan until you can see the actual mechanism that makes it true.

Step One: Learning to Destroy Images

Diffusion models are trained backwards from how you'd expect. Training starts with real images from a large captioned dataset, and a fixed, non-learned process called **forward diffusion**: a small amount of random (Gaussian) noise is added to the image, repeatedly, over many steps — a few hundred to a few thousand, depending on the model — until the image is statistically indistinguishable from pure noise. This forward process needs no learning at all; it's just repeated noise addition, and it's identical for every image in the dataset.

The interesting part — the part that's actually trained — is the **reverse** direction: a neural network is trained to look at a noisy image at some step and predict what noise was added, so that subtracting its prediction moves the image one step closer to clean. Do this once, and you've removed a little noise. Do it hundreds of times in sequence, starting from *pure* random noise instead of a slightly-noised real image, and the network — which has only ever seen "slightly less noisy version of this" as its training signal — ends up hallucinating a coherent image out of static, one small denoising step at a time.

1. **Training:** take a real image, add noise in increasing amounts across many steps, train a network to predict and remove the noise added at each step.

2. **Generation:** start from pure random noise, run the trained network's denoising step repeatedly, watch a coherent image emerge from static.

WHY THIS ALREADY EXPLAINS SOMETHING FROM IMGAI1-1

Notice what the network's training signal actually is at every single step: "given this noisy image (and, as covered next, this text), predict what noise was added." Never once, at any point in training, does the network see a signal

resembling "given this instruction, comply with it." The mechanism itself is the proof of imgai1-1's own claim, not just an assertion alongside it.

Step Two: Bringing Text Into the Process

Everything above describes an *unconditional* diffusion model — one that generates images with no text input at all, denoising toward whatever the training data made statistically likely in general. To make text prompting work, the denoising network needs the text to actually steer each denoising step, not just describe the final result after the fact.

This is where **CLIP** (Contrastive Language-Image Pretraining, an OpenAI model) or a similar text-image embedding model comes in. CLIP is trained on a separate, simpler task: given a large set of (image, caption) pairs, learn to map both images and text into the *same* numerical vector space, such that a genuinely matching image and caption land close together in that space, and mismatched pairs land far apart. The result is a text embedding — a long list of numbers — that captures, in a form the diffusion network can use, what "a photo of a golden retriever" means in visual terms.

During both training and generation, that text embedding is fed into the denoising network alongside the noisy image at every single step, biasing each prediction toward the region of "clean image space" the embedding points at. The prompt doesn't get read once and remembered — it's re-injected at every denoising step, continually pulling the emerging image toward the text's own embedding.

COMPONENT	JOB	TRAINED ON
CLIP (or similar)	Maps text and images into a shared vector space	(image, caption) pairs — a matching/mismatching task
Denoising network	Predicts and removes noise, steered by the text embedding	Progressively noised images, conditioned on their own caption's embedding

A Preview of Two Terms This Course Will Need Later

Two mechanical details, previewed here and covered in full technical depth in `imgai1-5`:

- **Latent diffusion:** running the denoising process directly on full-resolution pixels is expensive. Stable Diffusion instead compresses images into a smaller "latent" representation first (via a separate encoder network), runs the entire noise/denoise process there, and decompresses back to pixels only at the end — the same mechanism described above, just operating on a compressed representation rather than raw pixels.
- **Classifier-free guidance:** the network is actually trained to predict noise both *with* and *without* the text embedding, and generation blends the two predictions, amplifying the difference the text makes. Stable

Diffusion's own CFG scale parameter (`imgai1-5`) controls exactly how strongly that difference gets amplified.

Why This Mechanism Predicts Real, Later Failure Modes

NO STRUCTURAL UNDERSTANDING WAS EVER LEARNED

At no point in this entire process does the network learn a rule like "hands have five fingers" or "text is made of discrete, spellable characters." It only ever learns statistical pixel-arrangement patterns associated with embeddings. Hands are genuinely hard for this mechanism — they're small, highly variable, articulated structures that appear in countless different poses across training images, so the statistical pattern is much fuzzier than something like "a face." Rendering legible text is harder still — nothing in the training signal ever represents letters as discrete symbols with a required exact sequence, only as visual textures that loosely correlate with certain image regions. `imgai1-8` covers both of these failure modes in full, and traces each one back to this exact explanation rather than treating them as unexplained quirks.

Hands-On Exercises

EXERCISE 1

Using this chapter's own two-step description (forward diffusion, then the trained reverse process), explain why generation is able to start from pure random noise and still produce a coherent image, even though the network was only ever trained to remove small amounts of noise at a time.

View solution

EXERCISE 2

Explain what CLIP (or a similar model) actually does, what it's trained on, and why re-injecting the text embedding at every denoising step (rather than reading the prompt once at the start) matters for how strongly the text actually steers the result.

View solution

EXERCISE 3

Using this chapter's own warn-box, explain mechanically (not just "AI is imperfect") why hands and legible text are both genuinely hard for a diffusion model, tracing each difficulty back to what the network's training signal actually was.

View solution

CHAPTER 2 QUICK REFERENCE

- **Forward diffusion** (fixed, not learned) — repeatedly add noise to a real image until it's statistically pure noise
- **Reverse diffusion** (trained) — a network predicts and removes noise step by step; run from pure noise, this hallucinates a coherent image
- **CLIP-style embedding** — maps text and images into one shared vector space; the text embedding steers every denoising step, re-injected each time
- **Latent diffusion / CFG** previewed here, covered fully in `imgai1-5`
- No structural rules (finger counts, letter sequences) are ever learned — only statistical pixel patterns, which is why `imgai1-8`'s failure modes exist at all
- **Next chapter:** The Anatomy of an Effective Image Prompt

CHAPTER 3 OF 10

The Anatomy of an Effective Image Prompt

Generative AI Prompting for Image Models

Chapter 3 · The Anatomy of an Effective Image Prompt

`prompt1-2` built its own anatomy of a good prompt around four instruction-oriented pillars: Clarity, Context, Constraints, Output Format. That anatomy makes sense for a system built to comply with a request (`imgai1-1`). An image model needs a genuinely different anatomy — one built around *description*, not instruction — because, per `imgai1-2` 's own mechanism, every word in the prompt is just another statistical anchor pulling the denoising process toward a region of image space, over and over, at every step. This chapter builds that anatomy: six descriptor categories every tool-specific chapter from here on (`imgai1-4` through `imgai1-6`) assumes without re-teaching.

Six Categories, One Shared Vocabulary

Regardless of which of the three tools this course covers, a strong prompt tends to touch some combination of the same six things. Syntax differs wildly by tool (covered next, in `imgai1-4` – `imgai1-6`) — this vocabulary doesn't.

1

Subject

Who or what, doing what, with what specific details. The single biggest lever — vague subjects produce generic results (see this chapter's own warn-box).

`a scruffy terrier mid-jump` `catching a red frisbee` `weathered hands`

2

Style

The overall aesthetic or art-historical reference point. Named living artists raise real, direct questions — previewed here, covered fully in `imgai1-9` .

`cyberpunk` `art nouveau` `photorealistic` `watercolor`

3

Composition

Framing and layout — how the subject sits inside the frame.

`close-up` `wide shot` `rule of thirds` `symmetrical` `bird's-eye view`

4

Lighting

Where light comes from and how it feels — one of the most reliable levers for mood.

`golden hour` `dramatic rim lighting` `soft diffused light` `neon glow`

5

Camera / Lens

Real photography terminology — works precisely because it's what real photographs are captioned with (`imgai1-2`'s own CLIP training data).

`35mm lens` `shallow depth of field` `bokeh` `long exposure`

6

Medium

What kind of object the output should look like it is — the single strongest lever for realism vs. illustration.

`oil painting` `3D render` `pencil sketch` `digital art` `photograph`

WHY CAMERA/LENS LANGUAGE ACTUALLY WORKS

This isn't superstition. Per `imgai1-2`, CLIP was trained on real (image, caption) pairs scraped at massive scale — a huge fraction of which are real photographs with real photography metadata and captions attached ("shot on 35mm," "shallow depth of field"). The embedding space genuinely encodes what those terms visually correlate with, because real photographers' own captions taught it to. This is also exactly why the same terms don't work identically across all six categories equally well — camera language is unusually well-represented in the training data specifically because photography captioning is common online.

A Worked Before/After Example

BEFORE — SUBJECT ONLY

`a lighthouse`

AFTER — ALL SIX CATEGORIES

```
a weathered stone lighthouse on a rocky cliff [subject], oil painting [medium], impressionist style [style], wide shot with the lighthouse off-center [composition], dramatic golden-hour lighting with long shadows [lighting], shallow depth of field, 85mm lens [camera/lens]
```

Notice the "after" version doesn't read as an instruction to a listener — it reads as a dense list of descriptors, exactly as `imgai1-1` predicted. Nothing in it asks the model to do anything; every phrase describes what the finished image should look like.

WHY "A LIGHTHOUSE" ALONE PRODUCES SOMETHING GENERIC

Per `imgai1-2`'s own mechanism, the text embedding for a bare, vague prompt sits in a huge, poorly-differentiated region of the model's learned space — it's consistent with an enormous number of very different training images (photos, paintings, cartoons, day, night, close-up, distant). The denoising process gets pulled toward whatever's statistically most common across that whole broad region — which is exactly what "generic" means here: not a flaw in the output, but the predictable result of an under-specified target region. Every one of the six categories above exists specifically to narrow that region down.

Ordering Carries Real Weight

Across most tools, terms earlier in a prompt tend to exert more influence on the final image than terms later in it — not a hard universal rule, but a consistent enough pattern to plan around. This is one reason subject conventionally comes first: it's the single most important thing to get right, so it goes where it has the most leverage. `imgai1-7` covers the more precise, tool-specific mechanisms for controlling relative influence (term weighting) directly.

Hands-On Exercises

EXERCISE 1

Contrast this chapter's own six-category anatomy against prompt1-2's Clarity/Context/Constraints/Output Format anatomy, and explain — using imgai1-1's own distinction — why an image model needs a genuinely different anatomy rather than a relabeled version of the same one.

View solution

EXERCISE 2

Using this chapter's own warn-box and imgai1-2's own mechanism, explain mechanically why a vague, single-word prompt like "a lighthouse" produces a generic result, rather than describing it as simply "the AI being lazy."

View solution

EXERCISE 3

Explain, using this chapter's own tip-box, why real camera/lens terminology ("35mm lens," "shallow depth of field") is a genuinely effective descriptor category rather than an arbitrary one, tying the explanation back to imgai1-2's own description of CLIP's training data.

View solution

CHAPTER 3 QUICK REFERENCE

- Six shared descriptor categories: **Subject** · **Style** · **Composition** · **Lighting** · **Camera/Lens** · **Medium**
- Vague prompts produce generic results because their text embedding sits in a broad, poorly-differentiated region of the model's learned space (imgai1-2)
- Camera/lens language works because real photo captions are heavily represented in CLIP's own training data
- Earlier terms generally carry more influence — subject conventionally leads for this reason; imgai1-7 covers precise weighting mechanisms

- This vocabulary is shared across tools — syntax differs, starting with **Next chapter:** Midjourney — Parameters & Its Own Artistic Bias

CHAPTER 4 OF 10

Midjourney — Parameters & Its Own Artistic Bias

Generative AI Prompting for Image Models

Chapter 4 · Midjourney — Parameters & Its Own Artistic Bias

`imgai1-3`'s six-category vocabulary is shared across every tool in this course — but how that vocabulary is delivered to the model, and what the model does with it once it has it, differs a great deal by tool. This chapter is the first of three (`imgai1-4`–`imgai1-6`) covering that tool-specific layer, starting with the one running the most cloud-only, most curated, most parameter-driven of the three.

The Discord-Based Workflow

Midjourney has no standalone desktop app and no local install — it runs entirely in the cloud, and (traditionally) was accessed through Discord: joining Midjourney's own Discord server (or inviting its bot to a private server), typing `/imagine` followed by a prompt in a text channel, and receiving a 2×2 grid of four candidate images a short time later. A newer web-based interface now exists alongside Discord, but the underlying workflow is the same: submit a prompt, get four options, then either upscale one, generate variations of one, or rerun the whole prompt.

> **MIDJOURNEY IS CLOSED-SOURCE — THIS CHAPTER DESCRIBES DOCUMENTED BEHAVIOR, NOT INTERNALS**
>
> Unlike Stable Diffusion (`imgai1-5`), Midjourney doesn't publish its model architecture or training details. Everything in this chapter about what a parameter "does" reflects Midjourney's own published documentation and consistent, widely observed community behavior — not an academic description of internal mechanics the way `imgai1-2`'s diffusion explainer could be for the general case. Keep that distinction in mind: this chapter describes what Midjourney reliably *does*, not exactly *how* it does it internally.

Parameters — Appended, Not Woven In

Midjourney prompts follow a consistent shape: the descriptive prompt itself (built from `imgai1-3`'s six categories), followed by one or more `--parameter` flags appended at the end. Parameters configure the generation process itself rather than describing the image's content.

```
a weathered lighthouse on a rocky cliff, oil painting, dramatic golden-hour lighting --ar 16:9 --stylize 250 --chaos 15
```

PARAMETER	CONTROLS	NOTES
--ar (aspect ratio)	Output image proportions	e.g. `--ar 16:9` for widescreen, `--ar 1:1` for square (the default)
--stylize / --s	How strongly Midjourney's own trained aesthetic bias is applied vs. literal prompt adherence	Range roughly 0–1000; low values track the prompt more literally, high values lean into Midjourney's own house style
--chaos / --c	How much the four initial grid results vary from each other	Range roughly 0–100; low values produce four similar takes, high values produce four genuinely different interpretations
--no	Excludes specified content — Midjourney's own negative-prompt mechanism	`--no text, watermark`; the tool-specific implementation of the negative-prompt concept covered generally in `imgai1-7`
--iw (image weight)	How strongly an attached reference image influences the result relative to the text prompt	Used alongside an image URL provided directly in the prompt itself

--STYLIZE IS THE PARAMETER MOST WORTH UNDERSTANDING EARLY

A prompt that reads as extremely literal and technical ("a red cube, 3 inches, on a white background") at `--stylize 0` tends to produce something close to a literal, undecorated rendering of that description. The same prompt at `--stylize 750` often comes back noticeably more dramatic, atmospheric, and painterly than the prompt itself asked for — extra lighting, extra mood, extra visual polish the text never requested. This is the mechanism behind this chapter's next section.

Midjourney's Own Well-Documented Artistic Bias

Midjourney is widely and consistently observed — by its own users, in its own documentation, and across independent comparisons — to lean toward a dramatic, painterly, highly polished aesthetic even when a prompt doesn't ask for one. Flat, neutral, purely documentary-style requests often come back with cinematic lighting, rich color grading, and a level of visual "finish" the prompt never specified. This isn't a bug or an inconsistency; it's a real, deliberate product characteristic, widely understood to result from how Midjourney curates and fine-tunes its own model toward outputs its own community and internal review consistently rate as more visually striking.

This matters directly for how you write Midjourney prompts: getting a genuinely neutral, unstylized result on purpose usually takes deliberate, explicit effort (a low `--stylize` value, explicit style-suppressing language) rather than simply omitting style descriptors and expecting a blank slate.

	MIDJOURNEY	WHAT THIS SETS UP
Default tendency	Dramatic, painterly, highly polished, even unrequested	Contrasted with Stable Diffusion's more neutral baseline (imgai1-5) and DALL-E's own different tendency (imgai1-6)

	MIDJOURNEY	WHAT THIS SETS UP
Access model	Cloud-only, Discord/web, no local install	Contrasted with Stable Diffusion's own local-hosting capability (imgai1-5)
Openness	Closed-source, documented behaviorally	Contrasted with Stable Diffusion's own fully open architecture (imgai1-5)

Hands-On Exercises

EXERCISE 1

Explain what --stylize controls, using this chapter's own tip-box example (the "red cube" prompt at --stylize 0 vs. --stylize 750), and explain why the difference between those two outputs is a real, deliberate product characteristic rather than random inconsistency.

View solution

EXERCISE 2

Using this chapter's own warn-box, explain why this chapter can only describe what Midjourney's parameters do, not exactly how they work internally, and contrast this with how imgai1-2 was able to describe diffusion models in general.

View solution

EXERCISE 3

Explain the practical consequence of Midjourney's own artistic bias for someone trying to get a genuinely neutral, undecorated result — what do they actually have to do, and why doesn't simply omitting style descriptors work the way it might on a more neutral tool?

View solution

CHAPTER 4 QUICK REFERENCE

- **Workflow:** Discord (or web) — `/imagine` a prompt, get a 2×2 grid, upscale or vary
- **--ar** aspect ratio · **--stylize** literal-vs-house-style · **--chaos** variation across the initial grid · **--no** exclusion (tool-specific negative prompt) · **--iw** reference-image weight
- Midjourney has a real, documented bias toward dramatic/painterly results, even unrequested — getting neutral output takes deliberate effort
- Closed-source — this chapter describes documented behavior, not internal architecture

- **Next chapter:** Stable Diffusion — Open-Source Control & Technical Parameters

CHAPTER 5 OF 10

Stable Diffusion — Open-Source Control & Technical Parameters

Generative AI Prompting for Image Models

Chapter 5 · Stable Diffusion — Open-Source Control & Technical Parameters

`imgai1-4`'s own warn-box drew a real line: Midjourney's parameters can only be described by their observed behavior, because its architecture is never published. Stable Diffusion is the other side of that line. Its model weights, training methodology, and source code are all public — which means this chapter can do something `imgai1-4` explicitly couldn't: explain exactly what its own parameters do to the mechanism `imgai1-2` already introduced, not just what they're observed to produce.

Local, Self-Hosted, and Genuinely Open

Where Midjourney is cloud-only with no local install (`imgai1-4`), Stable Diffusion's model weights can be downloaded and run entirely on your own hardware — no account, no subscription, no internet connection required once set up. In practice, most people run it through a community-built interface (Automatic1111's web UI, ComfyUI, or InvokeAI are the most common) rather than writing raw code against the model directly, but the defining fact remains: nothing about running it requires anyone else's server.

> **THE REAL TRADE-OFF, STATED PLAINLY**
>
> Local hosting trades Midjourney's zero-setup convenience for real requirements: a GPU with enough VRAM to hold the model, and enough technical comfort to install and configure one of the community interfaces above. Hosted, pay-per-use Stable Diffusion services exist too, splitting the difference — but the genuinely local, self-hosted option is the thing no other tool in this course offers at all.

CFG Scale — Now Explainable in Full

`imgai1-2` previewed classifier-free guidance in one sentence: the network is trained to predict noise both with and without the text embedding, and generation blends the two predictions. Here's the full picture. At every denoising step, the model computes two separate noise predictions from the same noisy image: an **unconditional** prediction (as if no text prompt existed at all) and a **conditional** prediction (steered by the text embedding, per `imgai1-2`'s own CLIP mechanism). The final prediction actually used is the unconditional one, plus the *difference* between the two, amplified by the CFG scale:

```
final_prediction = unconditional_prediction + CFG_scale × (conditional_prediction − unconditional_prediction)
```

A CFG scale of 1 uses the conditional prediction roughly as-is. Higher values (commonly 7–12 as a working range) amplify the direction the text embedding is already pulling toward, producing results that adhere to the prompt more strongly. Pushed too high, that amplification overshoots — oversaturated colors, harsh contrast, and visible artifacting, sometimes called "overcooked" output. Lower values produce looser, more creative departures from the literal prompt, at the cost of adherence.

Sampling Steps & Samplers

Steps is simply how many times the reverse denoising loop (`imgai1-2`) actually runs — more steps generally mean a more fully resolved image, with sharply diminishing returns past a certain point (often somewhere around 20–50, depending on the sampler). **Samplers** (Euler, DPM++, DDIM, and others) are different numerical algorithms for solving that same denoising process — they don't change what the model has learned, only how efficiently and in what character it converges toward a final image. Different samplers can produce visibly different results from an identical prompt and seed, and some converge acceptably in far fewer steps than others.

Negative Prompts as a First-Class Mechanism

`imgai1-4` covered Midjourney's own `--no` parameter as a simple exclusion list. Stable Diffusion's negative prompt is a genuinely deeper mechanism, and the CFG formula above explains exactly why: instead of using a truly "unconditional" prediction (as if no text existed) as the baseline, a negative prompt is itself run through CLIP the same way the positive prompt is, and its own embedding *replaces* the unconditional prediction in the formula:

```
final_prediction = negative_prediction + CFG_scale × (positive_prediction − negative_prediction)
```

This doesn't just avoid the negative prompt's content — it actively steers generation *away* from that region of embedding space, at every single denoising step, amplified by the same CFG scale governing how strongly the positive prompt pulls toward its own target. A negative prompt is treated by the mechanism as a full, first-class second prompt, not a simple exclusion filter layered on afterward.

WHY THIS IS WORTH UNDERSTANDING, NOT JUST USING

Because the negative prompt shares the exact same CFG amplification as the positive one, an overly aggressive or overly broad negative prompt can distort a result just as much as an overly high CFG scale can — it isn't a free, side-

effect-free way to remove unwanted content. `imgai1-7` covers negative-prompt technique in more practical depth across tools.

Checkpoints & LoRAs — A Genuinely Unique Capability

Because Stable Diffusion's weights are public, the community has trained thousands of alternate **checkpoints** — full, re-trained or fine-tuned versions of the base model, often specialized toward a particular art style, subject matter, or level of photorealism. Swapping checkpoints changes the model's own underlying visual "instincts" before a single prompt word is even considered. **LoRAs** (Low-Rank Adaptation) are smaller, lighter-weight adapter files that can be layered on top of a checkpoint to nudge it toward a narrower style, character, or concept, without the cost of training or storing an entirely new full model. Neither of these has a real equivalent on Midjourney or DALL-E (`imgai1-6`) — both are closed, single-model, hosted-only systems with nothing analogous to swap in.

	MIDJOURNEY (IMGAI1-4)	STABLE DIFFUSION
Architecture	Closed — described behaviorally	Open — described mechanically, via imgai1-2
Hosting	Cloud-only	Local (self-hosted) or hosted
Negative prompts	--no, a simple exclusion parameter	A full second prompt, replacing the unconditional CFG baseline
Model customization	None — one fixed model	Checkpoints and LoRAs — swappable, layerable

Hands-On Exercises

EXERCISE 1

Using this chapter's own CFG formula, explain why a CFG scale of 1 behaves roughly like using the conditional prediction alone, and explain mechanically why pushing the scale too high produces oversaturated, "overcooked" results rather than simply "more accurate" ones.

View solution

EXERCISE 2

Explain, using this chapter's own two CFG formulas, exactly what changes when a negative prompt is added, and why this makes a negative prompt a "full, first-class second prompt" rather than a simple exclusion filter.

View solution

EXERCISE 3

Explain why checkpoints and LoRAs have no real equivalent on Midjourney or DALL-E, tying your answer back to imgai1-4's own warn-box about Midjourney being closed-source.

View solution

CHAPTER 5 QUICK REFERENCE

- Open-source: architecture, weights, and training methodology are all public — described mechanically, not just behaviorally
- **CFG scale** — amplifies the (conditional − unconditional) difference; too high overshoots into artifacting
- **Steps** — how many denoising iterations run · **Samplers** — the numerical algorithm used, affecting convergence speed and character
- **Negative prompts** replace the unconditional baseline in the CFG formula — a full second prompt, not a simple filter
- **Checkpoints** (full alternate models) and **LoRAs** (lightweight adapters) — a genuinely unique customization capability
- **Next chapter:** DALL-E & ChatGPT-Integrated Generation — Natural-Language Prompting

CHAPTER 6 OF 10

DALL-E & ChatGPT-Integrated Generation — Natural-Language Prompting

Generative AI Prompting for Image Models

Chapter 6 · DALL-E & ChatGPT-Integrated Generation — Natural-Language Prompting

`imgai1-4` and `imgai1-5` both required learning a real syntax layer on top of `imgai1-3`'s shared vocabulary — bracketed `--parameters` for Midjourney, a CFG scale and sampler choice for Stable Diffusion. DALL-E, as used today, has none of that. There's no parameter list to learn. As `imgai1-1` predicted back at the start of this course, this is the one tool genuinely close to `prompt1`'s own conversational territory — but understanding exactly *why* requires being precise about what's actually converging, and what isn't.

How It's Actually Used

DALL-E is used today primarily through ChatGPT's own conversational interface (a standalone API also exists for developers, but the conversational route is how most people actually prompt it). You type a plain-English description — no brackets, no flags — directly into the chat, the same way you'd talk to Claude in `prompt1`.

The Detail Most Tutorials Skip: ChatGPT Rewrites Your Prompt

Here's the part worth being precise about. When you type a request into ChatGPT for an image, **ChatGPT itself — a real, instruction-following LLM — reads your message, and internally rewrites and expands it into a more detailed, descriptor-rich prompt** before that rewritten version is ever handed to the underlying image-generation model. Your own words are almost never the literal text the image model receives.

1. You type a short, conversational request: "*a cozy reading nook*"
2. ChatGPT (an instruction-following LLM) interprets that request and expands it into a detailed, descriptor-rich prompt — filling in subject, style, lighting, composition, along roughly `imgai1-3`'s own six categories, even though you never specified them
3. That expanded prompt — not your original sentence — is what actually conditions the underlying image model's own generation process

THIS IS NOT THE SAME AS "DALL-E BECAME AN INSTRUCTION-FOLLOWING IMAGE MODEL"

> It's tempting to read this as evidence that `imgai1-1`'s own central claim doesn't apply to DALL-E — that unlike Midjourney or Stable Diffusion, this one really does "understand" your request. That's not quite right, and the distinction matters. The underlying image-generation model itself is still, per `imgai1-2`'s own mechanism, a diffusion-family system with no representation of a request as a request. What changed is that an *entirely separate, genuinely instruction-following system* (ChatGPT) now sits in front of it, translating your conversational instruction into exactly the kind of dense, descriptive prompt `imgai1-3` teaches — automatically, on your behalf. The convergence with `prompt1`'s own territory is real, but it happens at the **interface layer**, not the **image-generation mechanism** itself. Nothing in `imgai1-2`'s explanation of diffusion models stops being true for DALL-E specifically — it's just been given a skilled translator standing in front of it.

Iterative, Conversational Refinement — A Genuine Convergence

This is where the convergence with `prompt1-6`'s own iterative refinement technique is real and direct, not just superficial. Because generation happens inside an ongoing chat, you can follow up with plain conversational corrections — "make the sky more dramatic," "remove the hat," "try it at night instead" — and ChatGPT interprets that follow-up in context, using the same instruction-following comprehension it uses for any other conversational task, then produces a new expanded prompt reflecting the change. Midjourney's own variation/rerun workflow (`imgai1-4`) and Stable Diffusion's own re-run-with-adjusted-parameters workflow (`imgai1-5`) both require you to reconstruct or edit the underlying prompt/parameters yourself. DALL-E's own refinement loop is the only one of the three where a genuinely comprehending system is doing that reconstruction work for you.

What's Gained, and What's Genuinely Lost

	MIDJOURNEY / STABLE DIFFUSION	DALL-E (VIA CHATGPT)
Prompt style	Descriptor composition + tool-specific syntax	Plain conversational language
Refinement	Manual — edit prompt/parameters yourself, rerun	Conversational — describe the change, ChatGPT reconstructs the prompt
Granular control	CFG scale, samplers, seeds, checkpoints/LoRAs (imgai1-5)	None directly exposed — ChatGPT's own rewriting decides the details
Accessibility	Real learning curve (imgai1-3–imgai1-5)	Minimal — works reasonably well from a first, vague sentence

The trade-off is exactly what you'd expect once the mechanism is understood: accessibility and natural iteration in exchange for the fine-grained, direct control `imgai1-5`'s own CFG scale, sampler choice, and checkpoints/LoRAs provide. There's no way to hand ChatGPT a specific CFG value or swap in a community checkpoint — that whole layer of control is delegated to ChatGPT's own rewriting step, which is a black box in the sense that you don't see the expanded prompt it actually generates.

A Practical Consequence for How You Prompt It

Because ChatGPT fills gaps in your description before generation, a short, vague prompt on DALL-E doesn't collapse into the same kind of "generic" result `imgai1-3`'s own warn-box predicted for Midjourney or Stable Diffusion — ChatGPT will typically add its own reasonable specifics (a plausible style, lighting, composition) rather than leaving the model to average over a huge, under-specified region on its own. This doesn't mean specificity stops mattering — a genuinely detailed request still steers the result more precisely than a vague one — but the cost of vagueness is softer here than on the other two tools, because a real comprehending system is doing the gap-filling instead of the diffusion model itself.

Hands-On Exercises

EXERCISE 1

Using this chapter's own three-step flow, explain what ChatGPT actually does to your prompt before the image model ever sees it, and why "your words are almost never the literal text the image model receives" is an important detail rather than a minor technicality.

View solution

EXERCISE 2

Using this chapter's own warn-box, explain precisely why "DALL-E converges with prompt1's territory" is true at the interface layer but not true at the image-generation mechanism layer — and why conflating the two would misread what actually changed.

View solution

EXERCISE 3

Explain, using this chapter's own comparison table and imgai1-3's own warn-box about vague prompts, why a short, vague DALL-E prompt doesn't produce the same kind of generic result a short, vague Midjourney or Stable Diffusion prompt does.

View solution

CHAPTER 6 QUICK REFERENCE

- No bracketed parameters or CFG scale to learn — plain conversational prompting via ChatGPT

- ChatGPT (an instruction-following LLM) rewrites/expands your prompt into a detailed descriptor-rich version before the image model ever sees it
- The convergence with prompt1's own territory is real, but at the **interface layer** — the underlying image model is still a diffusion system with no request representation (imgai1-2)
- Conversational follow-ups ("make the sky more dramatic") are a genuine convergence with prompt1-6's own iterative refinement
- Trade-off: accessibility and natural iteration, in exchange for imgai1-5's own granular control (CFG, samplers, checkpoints/LoRAs)
- **Next chapter:** Universal Techniques — Weighting, Negative Prompts & img2img

CHAPTER 7 OF 10

Universal Techniques — Weighting, Negative Prompts & img2img

Generative AI Prompting for Image Models

Chapter 7 · Universal Techniques — Weighting, Negative Prompts & img2img

`imgai1-4` – `imgai1-6` each covered one tool's own particular syntax. This chapter zooms back out to three techniques that recur — in some form — across most of them, despite genuinely different syntax: **weighting** individual terms, **excluding** content, and **starting from an existing image** rather than pure noise. Each section cross-references back to the tool-specific version already covered.

Term Weighting

A whole-prompt CFG scale (`imgai1-5`) controls how strongly the *entire* prompt's embedding pulls generation. Term weighting is the more granular version of the same idea — controlling how strongly one specific *word or phrase* contributes to that overall pull, relative to the rest of the prompt.

TOOL	SYNTAX	EXAMPLE
Stable Diffusion (imgai1-5)	Parenthetical numeric weight	`(red umbrella:1.4), rainy street`
Midjourney (imgai1-4)	Double-colon multipliers	`red umbrella::2 rainy street::1`
DALL-E (imgai1-6)	No direct syntax — described conversationally	"*make the umbrella much more prominent than the street*", left to ChatGPT's own rewriting step

WEIGHTING TOO AGGRESSIVELY CAUSES THE SAME PROBLEM AS TOO-HIGH CFG

A term pushed to an extreme weight distorts the same way an overall CFG scale pushed too high does (`imgai1-5`) — the network is being asked to amplify one piece of its own conditioning signal well past the range it was trained to handle sensibly, and the result degrades rather than simply "emphasizing more." `imgai1-8` covers this class of failure — and several others — in full.

Negative Prompts, Recapped Across Tools

`imgai1-4` and `imgai1-5` already covered this in tool-specific depth — this section is a deliberately short recap, not new material. The goal is the same everywhere (exclude unwanted content), but the depth of mechanism genuinely differs:

TOOL	MECHANISM
Midjourney	`--no` — a simple exclusion parameter (imgai1-4)
Stable Diffusion	A full second prompt replacing the unconditional CFG baseline (imgai1-5)
DALL-E	No direct field — expressed conversationally ("don't include X"), interpreted by ChatGPT's own rewriting step (imgai1-6)

img2img — Starting From an Image, Not Pure Noise

`imgai1-2` described generation as starting from pure random noise and denoising down to a coherent image. **img2img** changes the starting point: instead of pure noise, the process starts from a real, existing image with only a *partial* amount of noise added, then runs the same trained denoising steps from there. Because the starting point already resembles a real image rather than static, the result tends to preserve the original's overall composition and structure while the denoising process fills in whatever the new text prompt specifies.

The amount of noise added to the starting image — often exposed directly as a **denoising strength** parameter — controls the trade-off precisely: a low value keeps the result very close to the original image (only lightly re-touched by the prompt), while a high value approaches full, pure-noise generation, where the original image's influence becomes minimal.

TOOL	HOW IMG2IMG IS INVOKED
Stable Diffusion	Direct — upload a reference image plus a denoising-strength value
Midjourney	An image URL supplied alongside the prompt, weighted via `--iw` (imgai1-4)
DALL-E	Attach a reference image in the chat and describe the desired change conversationally (imgai1-6)

Inpainting & Outpainting — A Genuinely Distinct Capability

img2img re-generates an *entire* image, just anchored to a starting point. **Inpainting** is more surgical: a mask marks a specific region of an existing image (a bad hand, an unwanted object), and only that masked region is regenerated — the rest of the image is held fixed at every denoising step, rather than merely encouraged to stay similar. **Outpainting** runs the same masking idea in reverse, extending an image past its original borders and generating new content in the newly added space that plausibly continues what's already there.

Both are available, with different interfaces, across all three tools — Stable Diffusion exposes inpainting/outpainting directly as a masking tool in most community interfaces; Midjourney and DALL-E both offer a comparable region-select-and-regenerate workflow through their own editing interfaces, without exposing the underlying masking mechanism directly.

Hands-On Exercises

EXERCISE 1

Explain, using this chapter's own weighting section and imgai1-5's own CFG formula, why term weighting is described as "the more granular version" of a whole-prompt CFG scale rather than an unrelated technique.

View solution

EXERCISE 2

Using this chapter's own img2img section and imgai1-2's own description of generation from pure noise, explain what a low vs. a high denoising strength actually does to the balance between the original image and the new prompt.

View solution

EXERCISE 3

Explain the real structural difference this chapter draws between img2img and inpainting — why is inpainting described as "more surgical" rather than just "img2img with a smaller area"?

View solution

CHAPTER 7 QUICK REFERENCE

- **Term weighting** — per-term version of CFG scale; SD uses parenthetical numeric weights, Midjourney uses `::` multipliers, DALL-E has none directly (conversational instead)
- **Negative prompts** — same goal everywhere, different depth: Midjourney's `--no` (simple exclusion) vs. SD's full second-prompt mechanism (imgai1-5)
- **img2img** — start denoising from a real image with partial noise, not pure noise; denoising strength controls how much the original survives
- **Inpainting** — regenerate only a masked region, rest held fixed · **Outpainting** — extend an image's borders with new, plausible content
- Weighting/CFG pushed too far degrades output the same way — previewing `imgai1-8`'s failure modes

- **Next chapter:** Common Failure Modes & Honest Limitations

CHAPTER 8 OF 10

Common Failure Modes & Honest Limitations

Generative AI Prompting for Image Models

Chapter 8 · Common Failure Modes & Honest Limitations

`imgai1-2`'s own warn-box previewed two failure modes in a single paragraph: hands, and legible text. This chapter delivers the full treatment of both, adds a third (prompt bleeding), and holds every one of them to the same standard set back then — a mechanical explanation traced to what the network's training signal actually was, never a shrug toward "AI isn't perfect yet."

FAILURE MODE 1

Anatomical Errors

Extra or missing fingers, fused digits, extra limbs — hands are the canonical, most-documented example.

FAILURE MODE 2

Text-Rendering Failure

Gibberish or misspelled lettering, inconsistent character shapes, signage that looks textual but spells nothing.

FAILURE MODE 3

Prompt Bleeding

Attributes from one described subject leaking onto another — a "red car and blue house" that comes back with a red-trimmed house.

Anatomical Errors, In Full

`imgai1-2` already named the root cause: "hands... are small, highly variable, articulated structures that appear in countless different poses across training images, so the statistical pattern is much fuzzier than something like 'a face.'" There is no rule anywhere in the network's training resembling "exactly five digits, each with a fixed number of joints" — only an averaged statistical impression of what hand-shaped regions of pixels tend to look like, blurred across an enormous range of real poses, angles, and partial occlusions.

Newer models have genuinely gotten better at this — larger, higher-resolution training sets and dedicated fine-tuning specifically targeting hands have measurably reduced how often extra or fused fingers appear. It's worth being precise about what that improvement actually is: a better-fitted statistical pattern, not a newly learned structural rule. The underlying mechanism hasn't changed; the average has simply gotten sharper.

WHY WEIGHTING A HAND DESCRIPTOR DOESN'T FIX ANATOMICAL ACCURACY

`imgai1-7` covered term weighting as amplifying how strongly a phrase's own embedding pulls the denoising process. Weighting `(detailed hands:1.5)` amplifies the pull toward whatever statistical pattern the model already associates with "hands" — it does not add a counting rule the model never learned in the first place. Amplifying a fuzzy average produces a more emphatically fuzzy average, not a precise one.

Text Rendering, In Full

`imgai1-2`'s own explanation: "nothing in the training signal ever represents letters as discrete symbols with a required exact sequence, only as visual textures that loosely correlate with certain image regions." A word is, to the network, a texture that tends to co-occur with signage, book covers, and similar contexts — not a sequence of discrete, individually meaningful characters that must appear in a specific, correct order.

Some newer tools have made real, visible progress here — but, honestly, mostly by adding something extra rather than by the same diffusion mechanism simply improving with scale: dedicated architectural components specifically for rendering legible text, or specialized training passes focused on typography, layered on top of the general diffusion process rather than emerging naturally from it. This is a meaningfully different kind of improvement from the hands case above — less "the same statistical pattern got sharper," more "a second, specialized mechanism was bolted on for this one problem."

Prompt Bleeding & Concept Mixing

A prompt describing two distinct subjects with distinct attributes — `a red car and a blue house` — doesn't guarantee those attributes stay cleanly attached to their intended nouns. The result might show a car with blue trim, or a house with a red door, or some blend of both. This is **prompt bleeding**, and it has its own distinct mechanical cause.

Per `imgai1-2`, the entire prompt is processed into a conditioning signal that steers the denoising network — but nothing in that process works like a strict grammatical parser that hard-binds each adjective to exactly the noun a human reader would assign it to. The network's own internal attention mechanism (how it decides which parts of the text embedding influence which regions of the developing image) is itself a learned, statistical association, not a rule-based binding — so during the many steps of denoising, a color or style term can end up influencing a region of the image other than the one it was "meant" for, especially when multiple similar subjects or attributes compete for the same visual region.

WHY THIS CONNECTS BACK TO IMGAI1-7'S OWN TECHNIQUES

This is exactly why `imgai1-7`'s weighting syntax, separated into distinct emphasized clauses, and more advanced regional-prompting features (assigning different prompts to different areas of the canvas directly, available in some Stable Diffusion interfaces) exist as practical mitigations — not fixes to the underlying mechanism, but ways of giving the attention process fewer opportunities to mix up which term belongs where.

What "Getting Better" Actually Means Here

IMPROVEMENT IS REAL; IT ISN'T THE SAME AS THE PROBLEM DISAPPEARING

None of these three failure modes is a simple bug scheduled to be patched away in the next release. Each one is a structural consequence of what a diffusion model actually learns — statistical pixel patterns, not symbolic rules or hard grammatical bindings (`imgai1-2`). Real, measurable progress on all three keeps happening, through larger training data, better architectures, and dedicated fixes layered on top — but "less frequent and less severe" is a genuinely different claim from "solved," and it's worth keeping the two distinct rather than assuming next year's model has simply fixed the underlying issue outright.

Hands-On Exercises

EXERCISE 1

Using this chapter's own warn-box, explain precisely why amplifying a hand descriptor's weight (imgai1-7) doesn't fix anatomical accuracy, and explain the real distinction between "a sharper statistical average" and "a newly learned structural rule."

View solution

EXERCISE 2

Explain why this chapter describes recent progress on text rendering as a "meaningfully different kind of improvement" from recent progress on hands, using the chapter's own distinction between a sharper statistical pattern and a bolted-on specialized mechanism.

View solution

EXERCISE 3

Using this chapter's own explanation of the attention mechanism, explain mechanically why "a red car and a blue house" can produce a house with red trim, and explain why imgai1-7's weighting/separation techniques count as mitigations rather than fixes to the underlying cause.

View solution

CHAPTER 8 QUICK REFERENCE

- **Anatomical errors** (hands) — a fuzzy statistical average of highly variable poses, no counting rule ever learned; weighting amplifies the fuzziness, not accuracy

- **Text rendering** — letters are learned as visual texture, not discrete symbols; recent fixes mostly bolt on specialized components rather than the base mechanism improving alone
- **Prompt bleeding** — the attention mechanism binding text to image regions is learned/statistical, not a grammatical parser; imgai1-7's weighting/separation techniques mitigate, not eliminate, this
- All three are structural, not bugs — "improving" and "solved" are genuinely different claims
- **Next chapter:** Ethics, Copyright & Responsible Use

CHAPTER 9 OF 10

Ethics, Copyright & Responsible Use

Generative AI Prompting for Image Models

Chapter 9 · Ethics, Copyright & Responsible Use

This chapter is deliberately substantive, matching the seriousness this site has given ethics elsewhere — `pentest1-1`'s written-authorization precondition, `crypto1`'s own real case studies. Four genuinely distinct issues are covered here, not one blurred-together "AI ethics" concern — they have different causes, different degrees of legal settlement, and different people actually responsible for addressing them. Treating them as one issue would obscure exactly the distinctions that matter.

Issue 1: Training-Data Copyright — Genuinely Unresolved

ONGOING LITIGATION, NOT A SETTLED QUESTION

Image models are trained (`imgai1-2`) on billions of images scraped from across the internet, a large share of which are under copyright, gathered without explicit licensing from the individual rights holders. The central legal question — does training a model on copyrighted images constitute infringement, or is it transformative fair use? — is genuinely being litigated right now, not settled either direction.

Getty Images v. Stability AI is a real, notable case specifically because Getty's own complaint pointed to outputs that reproduced a recognizable, garbled version of Getty's own watermark — direct, visible evidence that specific training images had been memorized closely enough to leave a trace in generated output, not merely "influenced" the model in some diffuse statistical sense. **Andersen v. Stability AI** is a separate class action brought by a group of working artists raising the same underlying training-data question from a different angle. Both are genuinely unresolved as of this writing — this chapter states the real question being litigated rather than asserting a confident answer the law itself hasn't reached.

Issue 2: Living-Artist Style Mimicry — A Legal Question and an Ethical Question, Kept Separate

TWO DIFFERENT AXES, OFTEN CONFLATED

Prompting `in the style of [named living artist]` (`imgai1-3`'s own Style category) is real, common, documented practice. It raises a genuinely different question from Issue 1 above, and the two shouldn't

be collapsed together.

The legal question: under U.S. copyright law, a specific work is protected — a particular painting, a particular photograph — but a general *style* (a recognizable way of using color, brushwork, composition) generally is not. Mimicking a living artist's style, narrowly, is not the same legal category as reproducing one of their specific copyrighted works.

The ethical question is separate, and real regardless of the legal answer: a working artist's distinctive style is often their own economic livelihood and reputation — the thing clients specifically hire them for. Generating unlimited, uncompensated, unconsented content that competes directly with that artist's own commissioned work, using their own name as a literal prompt term, causes a real, documented economic harm even in cases where no specific copyrighted work was reproduced. Several tools, including Midjourney, have restricted or removed the ability to invoke specific living artists' names by name as a matter of policy — a real, documented response to exactly this concern, independent of how the unsettled legal question in Issue 1 eventually resolves.

WHY KEEPING THESE TWO AXES SEPARATE MATTERS

"Style isn't copyrightable, so it's fine" answers only the legal question — it says nothing about the separate, real economic harm to a specific working artist. "It harms artists, so it must be illegal" makes the opposite mistake, treating a real ethical concern as if it settles a legal question it doesn't actually resolve. Both halves of this section are true at once, and neither cancels the other out.

Issue 3: Deepfakes & Consent — The Clearest-Cut Case in This Chapter

DIRECT, IDENTIFIABLE HARM TO A SPECIFIC REAL PERSON

This is a different category from both issues above — not about training data or artistic style, but about generating a realistic, identifiable image of a specific real person without their consent. This is the clearest-cut ethical case in this chapter, with the least genuine ambiguity: non-consensual explicit imagery (a well-documented, serious harm, with dedicated legislation emerging specifically to address it in multiple jurisdictions), fabricated images of public figures placed in fabricated situations for political disinformation, and more mundane identity misuse all fall here.

Most major tools now maintain real content policies restricting the generation of photorealistic images of real, named individuals, with genuinely varying enforcement effectiveness across tools and over time. Unlike Issues 1 and 2, there's no real live legal or ethical debate over whether this category of harm is real — the open questions here are almost entirely about detection, enforcement, and legislative response, not about whether the underlying concern is legitimate.

Issue 4: Training-Data Bias Surfacing in Generated Output

A MECHANICAL CONSEQUENCE, NOT AN INVENTED ONE

A bare, unspecified prompt like `a doctor` or `a CEO` has, across multiple tools, been well-documented to default toward particular demographics far more consistently than real-world demographics for those roles would suggest. This isn't a value the model invented from nothing — per `imgai1-2`'s own mechanism, and directly extending `imgai1-3`'s own explanation of why vague prompts produce generic, averaged results, an unspecified prompt's embedding sits in a broad region shaped by whatever demographic patterns were statistically dominant in the captioned training images associated with that term. If historical stock photography and web imagery skewed a particular way for a given role, the model's own statistical average reflects that skew mechanically, whether or not anyone building the model intended it.

This mechanical explanation doesn't absolve model builders of responsibility — a company choosing what data to train on, and whether to intervene on documented bias afterward, is still making real choices with real consequences. Some companies have made deliberate interventions to diversify default outputs for certain prompts, with mixed and sometimes controversial results when those interventions have been applied inconsistently or without enough care for the actual prompt's own context.

Four Issues, Different Responsibility

ISSUE	STATUS	PRIMARILY WHOSE RESPONSIBILITY
Training-data copyright	Genuinely unresolved, active litigation	Model builders / dataset curators; courts
Living-artist style mimicry	Legally narrow, ethically real	Both platform policy and individual prompting choices
Deepfakes / consent	Clear-cut harm; open questions are about enforcement	Primarily the individual user; platform policy as a backstop
Training-data bias	Well-documented, mechanically explainable	Primarily model builders, via dataset and intervention choices

Hands-On Exercises

EXERCISE 1

Explain why Getty Images v. Stability AI is described in this chapter as genuinely notable evidence, specifically because of the watermark detail, rather than just "another lawsuit about training data."

View solution

EXERCISE 2

Using this chapter's own warn-box, explain why "style isn't copyrightable, so mimicking a living artist's style is fine" is an incomplete argument, and explain what question it actually answers versus what question it leaves untouched.

View solution

EXERCISE 3

Using this chapter's own explanation of training-data bias and imgai1-3's own mechanism for why vague prompts produce generic results, explain why a biased default output is described as "mechanical" rather than "invented," and explain why this doesn't remove responsibility from the companies that build these models.

View solution

CHAPTER 9 QUICK REFERENCE

- **Training-data copyright** — genuinely unresolved, active litigation (Getty v. Stability AI, Andersen v. Stability AI)
- **Living-artist style mimicry** — style itself isn't copyrightable (legal axis), but real economic harm to working artists is a separate, genuine ethical axis
- **Deepfakes/consent** — the clearest-cut harm in this chapter; open questions are about enforcement, not legitimacy
- **Training-data bias** — a mechanical consequence of skewed training data (imgai1-2/imgai1-3's own averaging mechanism), not an invented value, though builders remain responsible for it
- Four distinct issues, four different primary responsibility-holders — resist collapsing them into one blob
- **Next chapter:** Capstone: Crafting a Prompt Iteration Workflow

CHAPTER 10 OF 10

Capstone: Crafting a Prompt Iteration Workflow

Generative AI Prompting for Image Models

Chapter 10 · Capstone: Crafting a Prompt Iteration Workflow

One creative brief, taken through five real refinement passes, on one tool — Stable Diffusion (`imgai1-5`), chosen specifically because it's the only tool in this course whose mechanism (`imgai1-2`) can be reasoned about precisely at every step, rather than described only behaviorally (`imgai1-4`) or mediated through a rewriting layer (`imgai1-6`).

THE BRIEF

A promotional banner image for a fictional artisan coffee shop, **"Ember & Oak"**: a warm, inviting, photorealistic interior shot suitable for a website header, showing a barista at the counter with a hand-lettered menu board visible in the background.

Pass 1 — The Naive Prompt

Prompt

```
a coffee shop interior
```

⚠ Problem: exactly the failure imgai1-3's own warn-box predicted — five words map to a huge, poorly-differentiated region of embedding space. The result is a plausible but generic coffee shop, with no warmth, no barista, no signage, and no connection to "Ember & Oak" at all.

Pass 2 — Applying the Six-Category Vocabulary (imgai1-3)

Prompt

```
a cozy artisan coffee shop interior, a barista standing behind a wooden counter, hand-lettered chalkboard menu board on the wall behind [subject]; warm rustic photorealistic style [style]; wide shot, counter centered, chalkboard visible in background [composition]; warm golden
```

```
interior lighting, soft window light from the left [lighting]; 35mm lens, shallow depth of
field [camera/lens]; photograph [medium]
```

⚠ Problem: much stronger overall, but the chalkboard menu text comes back as garbled, illegible lettering — exactly the mechanical limitation imgai1-2 and imgai1-8 explained (letters are learned as visual texture, not discrete symbols). No amount of rewording the subject line fixes this on its own.

A DELIBERATE SCOPE DECISION, NOT AN OVERSIGHT

Per `imgai1-8` 's own honesty about text rendering, this is a structural limitation, not a wording bug. Rather than fighting it further, this workflow makes a deliberate choice: keep the chalkboard visually present as background texture (it still reads as "a menu board," which serves the brief), but don't attempt to force specific legible text onto it. Chasing perfectly legible generated text would require dedicated tools/techniques outside this chapter's own scope (see this chapter's closing scope note).

Pass 3 — Negative Prompt & CFG Adjustment (imgai1-5)

Negative prompt added

```
negative prompt: blurry, extra limbs, deformed hands, watermark, oversaturated, cartoon,
illustration
CFG scale: 8
```

⚠ Problem observed at this pass: the barista's hand near the espresso machine shows a fused-finger artifact — the anatomical failure mode imgai1-8 covered in depth.

✓ Per imgai1-5's own CFG formula, the negative prompt's own embedding replaces the unconditional baseline, actively steering away from "deformed hands" at every denoising step, in addition to the general quality terms — a first, partial mitigation.

Pass 4 — Term Weighting (imgai1-7)

Weighted subject clause

```
a cozy artisan coffee shop interior, a barista standing behind a wooden counter, (hands
resting on the counter, not visibly gripping anything:1.2), hand-lettered chalkboard menu
board on the wall behind ...
```

✓ Per imgai1-7's own weighting mechanism, this doesn't add an anatomical rule the model never learned (imgai1-8's own warn-box on this exact point) — it does, however, steer the pose itself toward a simpler, less articulated hand position, which reduces how often the fuzziest, highest-variability poses (the ones most prone to fused-finger artifacts) get generated in the first place.

Pass 5 — A Checkpoint Swap (imgai1-5)

Model change, not a prompt change

The base checkpoint's own default style leaned slightly more illustrative than the brief's "photorealistic" requirement wanted. Per `imgai1-5`, this is exactly the situation a checkpoint swap exists for: switching to a community checkpoint fine-tuned specifically toward photorealistic interior photography changes the model's own underlying visual instincts before a single prompt word is reconsidered, rather than trying to fight the base checkpoint's own bias with ever-more-specific style language.

A COPYRIGHT-CONSCIOUS STYLE CHOICE (IMGAI1-9)

Notice what Pass 2's own Style category deliberately does *not* do: it never names a specific living photographer or illustrator to imitate. Per `imgai1-9`'s own Issue 2, invoking a specific living artist's name raises a real, separate ethical question about economic harm, independent of the unsettled legal question in Issue 1 — generic style language ("warm rustic photorealistic style") sidesteps that question entirely while still achieving the brief's own goals.

Chapter Attribution

CAPSTONE ELEMENT	DRAWN FROM
Recognizing why Pass 1 failed	imgai1-3 (vague-prompt embedding-region mechanism)
Six-category prompt structure	imgai1-3 (Subject/Style/Composition/Lighting/Camera-Lens/Medium)
Choosing Stable Diffusion specifically	imgai1-2 (mechanism), imgai1-5 (explainable parameters)
Recognizing the chalkboard-text limitation	imgai1-2 / imgai1-8 (text-rendering mechanism)
Negative prompt & CFG scale	imgai1-5 (the full CFG/negative-prompt formula)
Recognizing the hand artifact honestly	imgai1-8 (anatomical-error mechanism)
Weighted pose clause	imgai1-7 (term weighting), applied with imgai1-8's own honest limits in mind
Checkpoint swap	imgai1-5 (checkpoints as a genuinely unique open-source capability)
Avoiding a named living artist	imgai1-9 (Issue 2 — style mimicry's separate ethical axis)

Honest Scope Note

WHAT THIS CAPSTONE DELIBERATELY DOESN'T ATTEMPT

- **One tool only.** This workflow was built and refined for Stable Diffusion specifically. Midjourney (`imgai1-4`) and DALL-E (`imgai1-6`) would each need their own tool-specific version of this same iteration process — the underlying six-category vocabulary (`imgai1-3`) transfers, the exact syntax and available controls don't.
- **No fine-tuning or LoRA-training walkthrough.** Pass 5 swaps to an existing community checkpoint — it doesn't cover how to train a new checkpoint or LoRA from scratch, a genuinely separate skill set beyond this course's own prompting focus.
- **No video-generation models.** This course, start to finish, covers still-image diffusion models only.
- The chalkboard text remains imperfect even after five passes — per `imgai1-8` 's own closing point, this is a real limitation that gets less severe with better tools, not one this workflow, or any prompting technique alone, fully eliminates.

Hands-On Exercises

EXERCISE 1

Explain why this capstone deliberately chose not to keep fighting the chalkboard's illegible text through further prompt rewording, using this chapter's own warn-box and imgai1-8's own honest distinction between mitigation and elimination.

View solution

EXERCISE 2

Explain the difference between what Pass 3's negative prompt fixes and what Pass 4's term weighting fixes for the hand-artifact problem, and why both were needed rather than either alone.

View solution

EXERCISE 3

Explain why this capstone was deliberately built on Stable Diffusion rather than Midjourney or DALL-E, using this chapter's own opening reasoning and the honest scope note's own admission about what would need to change for another tool.

View solution

CHAPTER 10 QUICK REFERENCE — COURSE SUMMARY

- Image prompting is descriptor composition, not instruction-giving (imgai1-1), grounded in a real diffusion mechanism (imgai1-2)
- Six shared descriptor categories (imgai1-3), wrapped in tool-specific syntax: Midjourney's parameters (imgai1-4), Stable Diffusion's explainable technical controls (imgai1-5), DALL-E's ChatGPT-mediated conversation (imgai1-6)
- Cross-tool techniques — weighting, negative prompts, img2img, inpainting (imgai1-7)
- Honest, mechanically-grounded limitations — anatomy, text, prompt bleeding (imgai1-8)
- Four distinct ethical issues, four different responsibility-holders (imgai1-9)
- This capstone combined all nine prior chapters into one real, five-pass iteration workflow