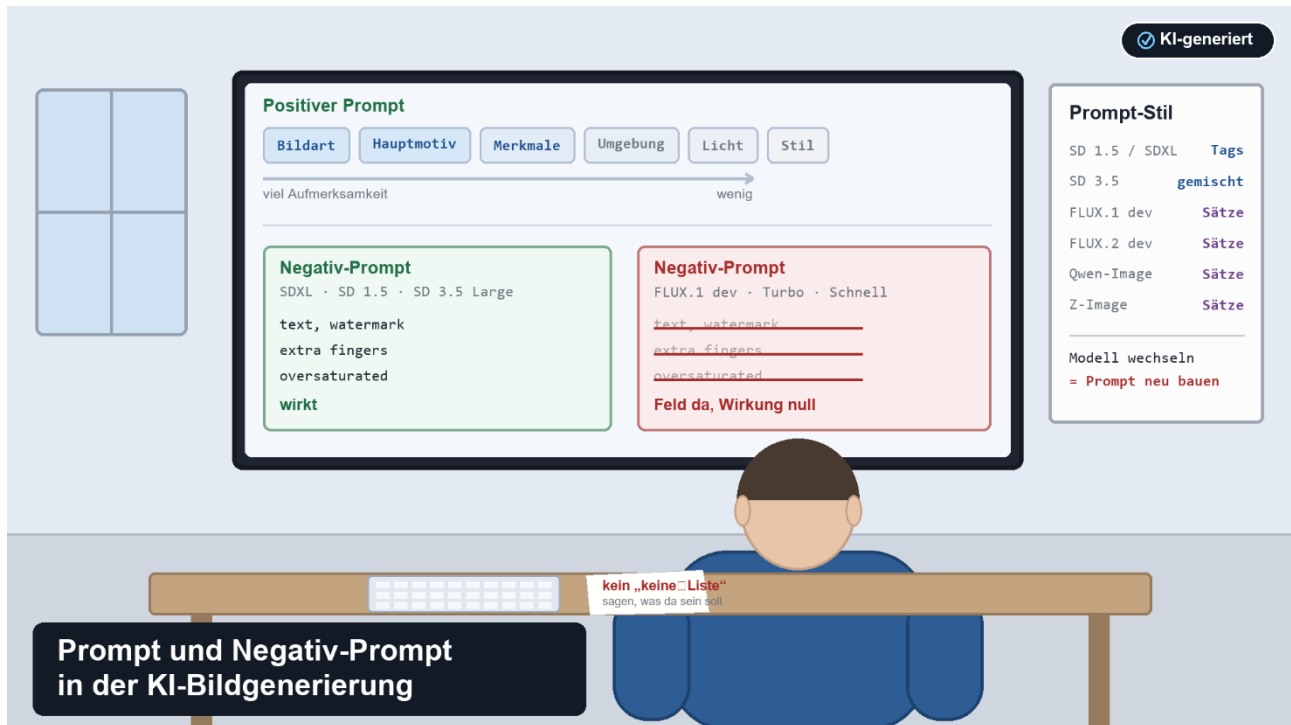


Prompt und Negativ-Prompt in der KI-Bildgenerierung



Links das Feld, das wirkt. Rechts dasselbe Feld an einem destillierten Modell – vorhanden, beschriftet, wirkungslos.

Wissen	Beratung	Schulung
Alle Grundlagen zur KI-Bildgenerierung an einer Stelle – vom Diffusionsprinzip bis zur Governance.	Aus Prompt-Basterei wird eine Bildproduktion, die auch in sechs Monaten dasselbe liefert.	Dein Team lernt, Prompts zu bauen statt zu raten – inklusive der Modelle ohne Negativ-Feld.
KI-Bildgenerierung	KI-Bildgenerierung Beratung	KI-Bildgenerierung Schulung

„Ein schönes Foto von unserer Produktion“ – und dann schaust du auf ein Bild, das aussieht wie eine Fabrikhalle aus einem Videospiel, in der jemand versehentlich sieben Finger an eine Hand gemalt hat. Der Prompt ist die einzige Eingabe, bei der du die vollständige Kontrolle hast, und gleichzeitig die, an der die meisten am sorglosesten arbeiten. Alle anderen Regler sind Zahlen; der Prompt ist die Bauanleitung.

Ich habe einmal eine Marketingabteilung übernommen, die über vier Wochen rund tausend Bilder erzeugt hatte, um zwölf brauchbare zu bekommen. Die Trefferquote lag unter anderthalb Prozent. Niemand hatte ein Problem mit der Hardware, mit dem Modell oder mit dem Budget. Das Problem war, dass jeder Versuch drei Dinge gleichzeitig änderte und niemand aufschrieb, welcher Prompt zu welchem Bild gehörte. Nach zwei Tagen Aufräumen lag die Quote bei knapp einem Drittel – ohne eine einzige neue Grafikkarte.

Die gute Nachricht: Du musst weder Dichter noch Mathematiker sein. Du musst nur wissen, wie dein Modell den Text überhaupt liest – und das hat sich in den letzten beiden Jahren grundlegend geändert. Wer heute noch Stichwortketten in ein FLUX.2 tippt, verschenkt drei Viertel dessen, was

das Modell kann. Und wer eine sorgfältig gepflegte Negativliste an ein destilliertes Modell schickt, hat einfach nur schön getippt.

Prompt und Negativ-Prompt in fünf Sätzen

Der positive Prompt beschreibt, was im Bild sein soll. Der Negativ-Prompt beschreibt, was nicht hineingehört – aber nur bei Modellen, die mit einem echten Guidance-Wert über 1 arbeiten. Die Gewichtung in Klammern, etwa (rote Jacke:1.3), stellt einzelne Bausteine lauter oder leiser; ihr Arbeitsbereich liegt zwischen 1,1 und 1,4. Ältere Modelle mit CLIP-Text-Encoder lesen Stichwortlisten in Blöcken von 75 Token; moderne Modelle mit großen Sprachmodellen als Text-Encoder verarbeiten bis zu 512 Token und wollen ganze Sätze. Welche Schreibweise und welches Negativ-Verfahren gilt, hängt also am Modell – nicht an deinem Geschmack.

Wie das Modell deinen Prompt überhaupt liest

Ein Bildmodell versteht deinen Text nicht selbst. Es gibt einen vorgeschalteten Text-Encoder, der aus deinem Prompt eine Zahlenwolke macht, und erst diese Zahlenwolke sieht das eigentliche Diffusionsmodell. Welcher Encoder das ist, entscheidet praktisch alles: wie lang dein Prompt sein darf, ob Grammatik hilft oder schadet, ob Gewichtungsklammern sauber funktionieren und wie viel räumliche Logik ankommt.

Vom Stichwort-Stapel zum ganzen Satz

Die erste Generation – Stable Diffusion 1.5 und SDXL, dazu die riesige Familie an Community-Ablegern wie Pony oder Illustrious – arbeitet mit CLIP als Text-Encoder. CLIP wurde darauf trainiert, Bilder und kurze Bildunterschriften einander zuzuordnen. Es versteht Konzepte, aber keine Erzählung. Ein Satz wie „die Katze sitzt links vom Hund“ zerfällt zu „Katze, Hund, links, sitzen“ – und wo die Katze am Ende sitzt, bleibt Glückssache. Deshalb sind Tag-Ketten mit Kommas hier tatsächlich die bessere Schreibweise: Sie täuschen keine Struktur vor, die ohnehin nicht ankommt.

Technisch verarbeitet CLIP Text in Blöcken von 75 Token plus zwei Steuerzeichen. Längere Prompts werden in mehrere solcher Blöcke zerlegt, die getrennt kodiert und anschließend aneinandergehängt werden. Das hat eine unangenehme Nebenwirkung: Eine Gewichtung wirkt nur innerhalb ihres Blocks, und Eigenschaften bluten über Blockgrenzen gern in benachbarte Motive aus. Wer in einer Oberfläche der Automatic1111-Familie arbeitet, kann mit dem Schlüsselwort BREAK – groß geschrieben – einen Block vorzeitig beenden und so verhindern, dass die rote Jacke auf den blauen Stuhl abfährt.

Die zweite Generation hat den Encoder ausgetauscht. FLUX.1 nutzt zusätzlich zu CLIP einen T5-Encoder, FLUX.2 [dev] arbeitet mit einem einzigen Text-Encoder auf Basis von Mistral Small 3.1 mit 24 Milliarden Parametern, die kleineren FLUX.2-[klein]-Varianten setzen auf Qwen3, und Qwen-Image verwendet das Vision-Language-Modell Qwen2.5-VL. Das sind ausgewachsene Sprachmodelle. Sie verstehen Nebensätze, Bedingungen, räumliche Beziehungen und Aufzählungen. Die maximale Prompt-Länge liegt bei diesen Modellen bei 512 Token – und im Unterschied zu CLIP ist das ein zusammenhängendes Fenster, kein Stapel aus Blöcken.

Der Bauplan eines Prompts

Fünf Bausteine, vom Wichtigsten zum Beiwerk – und zwei Schreibweisen für dieselbe Sache



Die Reihenfolge der Bausteine bleibt in beiden Schreibweisen dieselbe. Nur die Verpackung ändert sich.

*Die Reihenfolge der Bausteine bleibt in beiden Welten gleich. Was sich unterscheidet, ist die Verpackung:
 Kommaketten für CLIP, ganze Sätze für die großen Encoder.*

Der Bauplan, der in beiden Welten trägt

Unabhängig von der Schreibweise hat sich eine Reihenfolge bewährt, die du dir einmal einprägst und dann nicht mehr diskutierst: zuerst die Bildart, dann das Hauptmotiv, dann dessen Merkmale, dann die Umgebung, zuletzt Licht und Stil. Der Grund ist nicht Ästhetik, sondern Statistik: Was früh im Prompt steht, bekommt in allen gängigen Encodern tendenziell mehr Gewicht, und was am Ende steht, geht bei langen Prompts am ehesten unter.

„Bildart zuerst“ ist dabei der Baustein, den fast alle vergessen. Ob du „Produktfoto“, „technische Illustration“, „Bleistiftskizze“ oder „Screenshot“ schreibst, verschiebt das gesamte Ergebnis stärker als zehn Stilwörter am Ende. Das Modell hat diese Kategorien während des Trainings sauber gelernt, weil Bildunterschriften sie praktisch immer enthalten.

Modellfamilie	Text-Encoder	Prompt-Stil	Länge	Negativ-Prompt
SD 1.5 / SDXL und Ableger	CLIP	Tag-Ketten mit Kommas	75 Token je Block	ja, wirkt stark
SD 3.5 Large	CLIP plus T5	Tags und Sätze gemischt	bis 256 Token	ja
SD 3.5 Large Turbo	CLIP plus T5	kurze Sätze	bis 256 Token	nein, destilliert
FLUX.1 [dev] / [schnell]	CLIP plus T5	ganze Sätze	bis 512 Token	nein, Feld ohne Wirkung
FLUX.2 [dev]	Mistral Small 3.1	ausformulierte Beschreibung	bis 512 Token	nein, Guidance-Embedding
FLUX.2 [klein]	Qwen3	ausformulierte Beschreibung	bis 512 Token	Basis ja, Turbo nein
Qwen-Image	Qwen2.5-VL	ganze Sätze, auch mehrsprachig	lang	Feld vorhanden, kaum Wirkung
Z-Image	eigener LLM-Encoder	ganze Sätze	lang	je nach Variante

Der Zwei-Minuten-Test vor jedem neuen Modell

Bevor du eine Prompt-Bibliothek auf ein neues Modell umstellst, nimm einen einzigen Prompt und

schreibe ihn zweimal: einmal als Kommakette, einmal als ausformulierten Absatz. Gleicher Seed, gleiche Schrittzahl, gleicher Sampler. Nach vier Bildern weißt du, in welcher Welt du dich befindest. Das spart dir die nächsten zweihundert Fehlversuche.

Welche Modelle wie ticken, ordnet [Die 15 wichtigsten Modelle zur lokalen Bildgenerierung](#) ein; wer noch beim Auswählen ist, findet in der [Übersicht der wichtigsten KI-Bildgeneratoren](#) die Landkarte dazu.

Konkret schlägt blumig

Der häufigste Fehler in Unternehmens-Prompts ist nicht Länge, sondern Vagheit. „Schönes Licht“, „hochwertig“, „professionell“, „modern“ – das sind Wörter, die im Trainingsmaterial über Millionen völlig verschiedener Bilder verteilt sind. Sie bedeuten für das Modell praktisch nichts und verbrauchen trotzdem Aufmerksamkeit. „Weiches Fensterlicht von links, leichte Schatten nach rechts“ dagegen ist eine Ansage, die im Training eindeutig belegt ist.

Dasselbe gilt für Mengen und Positionen. „Mehrere Personen“ liefert dir zwischen zwei und neun Personen, je nach Laune des Seeds. „Drei Personen an einem Besprechungstisch, zwei sitzend, eine stehend“ ist bei Modellen mit großem Text-Encoder eine Anweisung, die tatsächlich ankommt – bei CLIP-Modellen dagegen nach wie vor ein Wunschzettel.

Gewichtung: einzelne Bausteine lauter stellen

Irgendwann passiert der Klassiker: Das Modell ignoriert ausgerechnet das eine Detail, auf das es dir ankommt. Der Helm ist blau, aber die Kratzer fehlen. Oder es übertreibt: Du wolltest ein bisschen Nebel und bekommst ein Bild, in dem man das Motiv suchen muss. Für beide Fälle gibt es die Gewichtung.

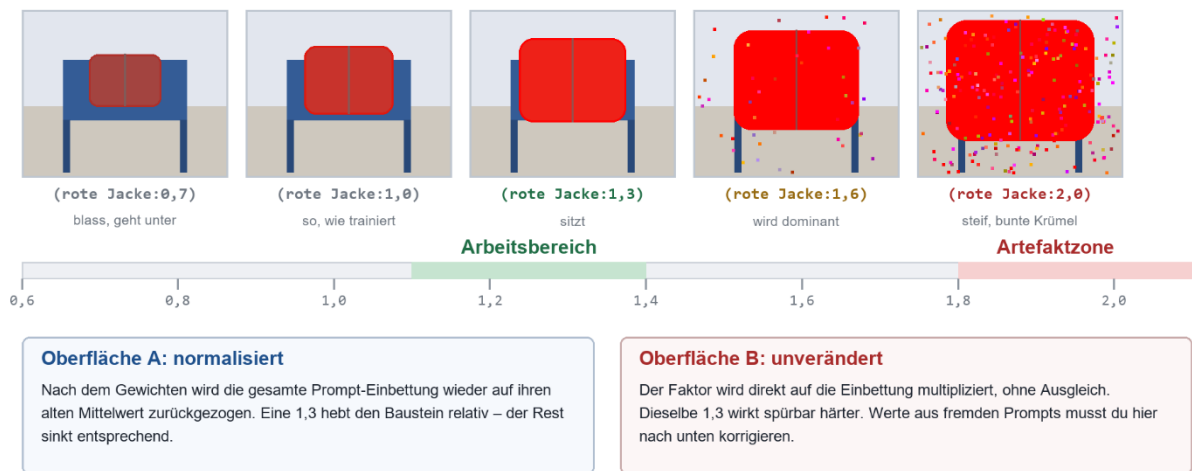
Die Klammer-Syntax und ihre Tücken

In den meisten Oberflächen schreibst du eine Gewichtung als Klammersausdruck mit Doppelpunkt und Zahl: (rote Jacke:1.3) betont, (rote Jacke:0.7) nimmt zurück. Der Wert 1.0 ist neutral. Technisch wird der entsprechende Abschnitt der Text-Einbettung mit diesem Faktor multipliziert, bevor er ins Modell geht.

Und genau hier beginnt die Verwirrung, die dich sonst Stunden kostet: Dieselbe Zahl bedeutet in zwei Programmen nicht dasselbe. Oberflächen der Automatic1111-Familie normalisieren nach dem Gewichten – die Gesamt-Einbettung wird auf ihren ursprünglichen Mittelwert zurückgezogen, sodass eine 1.3 den Baustein relativ zum Rest hebt. ComfyUI multipliziert in der Voreinstellung direkt, ohne Ausgleich. Ergebnis: Dieselbe 1.3 wirkt in ComfyUI deutlich härter. Wer Prompts aus einer Community-Sammlung übernimmt, sollte die Werte dort erst einmal um ein bis zwei Zehntel senken.

Gewichtung: (rote Jacke:x) in Wirkung übersetzt

Derselbe Prompt, derselbe Seed – nur die Zahl in der Klammer wandert



Merke: Eine Gewichtszahl ist kein Naturgesetz, sondern eine Angabe an eine bestimmte Oberfläche.

Von 0,7 bis 2,0 am selben Motiv. Bis etwa 1,4 arbeitest du, ab 1,8 wird das Bild steif und die Farben laufen aus dem Ruder.

Wie weit du drehen darfst

Der Arbeitsbereich liegt zwischen 1,1 und 1,4. Darunter merkst du kaum etwas, darüber wird es schnell unangenehm. Ab ungefähr 1,6 beginnt das Motiv das Bild zu dominieren, ab 1,8 kippen die Farben, Kanten bekommen Lichtsäume und im Hintergrund tauchen bunte Krümel auf. Das ist kein Fehler des Werkzeugs: Das Modell wurde nie auf derart aufgeblasene Einbettungen trainiert, also extrapoliert es – und Extrapolation sieht bei Diffusionsmodellen immer gleich hässlich aus.

Abschwächen funktioniert übrigens verlässlicher als Verstärken. Wenn dir etwas zu stark durchschlägt, ist (begriff:0.8) fast immer die sauberere Lösung als der Versuch, alles andere hochzuziehen. Das ist derselbe Reflex wie beim Mischpult: Erst die zu lauten Spuren runter, dann erst den Gesamtpegel anfassen.

Gewichtung ist kein Ersatz für einen sauberen Prompt

Wenn du drei oder mehr Gewichtungsklammern brauchst, um ein Bild in den Griff zu bekommen, ist nicht die Gewichtung das Problem, sondern der Grund-Prompt. Meistens stecken Widersprüche darin – „Frontalansicht“ und „Blick über die Schulter“, „Weitwinkel“ und „starke Hintergrundunschärfe“. Das Modell kann sich nicht entscheiden und liefert einen Kompromiss, den niemand bestellt hat. Erst aufräumen, dann feilen.

Wo Gewichtung kaum noch ankommt

Bei Modellen mit großem Sprachmodell als Text-Encoder verliert die Klammer-Syntax an Schärfe. Die Einbettung eines Mistral- oder Qwen3-Encoders ist kein simpler Token-für-Token-Vektor mehr, auf den man punktuell einen Faktor multiplizieren könnte, ohne den Satzzusammenhang zu beschädigen. Die Oberflächen bieten die Klammern weiterhin an, und sie tun auch irgendetwas – nur eben unschärfer und mit schnellerem Umkippen ins Artefakthafte.

Die bessere Antwort bei diesen Modellen ist Sprache. Statt (Kratzer:1.4) schreibst du „deutlich sichtbare Kratzer im Lack, an den Kanten bis aufs Metall durchgescheuert“. Ein Sprachmodell-

Encoder belohnt Präzision in Worten, nicht Faktoren in Klammern. Das ist gewöhnungsbedürftig, wenn man jahrelang mit Tag-Ketten gearbeitet hat, aber es ist die deutlich robustere Methode.

Werkzeug	Schreibweise	Besonderheit	Startwert
Automatic1111 / Forge	(begriff:1.3)	normalisiert nach dem Gewichten; BREAK verfügbar	1,2
ComfyUI	(begriff:1.3)	multipliziert direkt, wirkt härter	1,1
InvokeAI	begriff+ / begriff-	Plus- und Minuszeichen stapelbar	ein Plus
Foocus	(begriff:1.3)	zusätzlich eigene Stil-Voreinstellungen	1,2
Diffusers-Code	kein eigenes Format	Gewichtung nur über Zusatzbibliotheken	entfällt

Wie stark das Modell dem Prompt insgesamt folgt, regelt nicht die Gewichtung, sondern ein globaler Wert – das ist ein eigenes Thema und in [CFG-Scale: Guidance in KI-Bildgeneratoren](#) ausführlich erklärt. Die beiden hängen zusammen: Bei hoher Guidance wirken Gewichtungen härter, bei niedriger verpuffen sie. Wer beide gleichzeitig verstellt, sucht anschließend lange.

Der Negativ-Prompt und wie er tatsächlich arbeitet

Bei klassischen Stable-Diffusion-Modellen gibt es neben dem positiven Prompt ein zweites Textfeld. Da hinein kommt, was im Bild nicht vorkommen soll: Text, Wasserzeichen, verunglückte Hände, überzogene Farben, Rahmen, Signaturen. Der Negativ-Prompt ist eines der wirksamsten Werkzeuge der ganzen Disziplin – und wird in der Praxis erstaunlich oft leer gelassen.

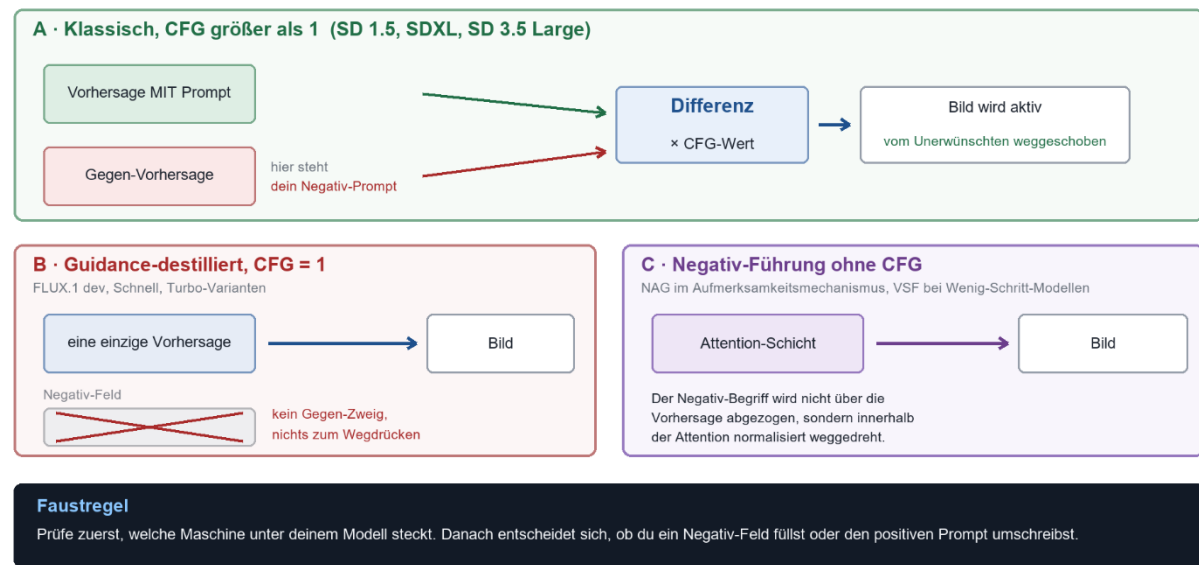
Der Mechanismus in einem Absatz

Ein klassisches Diffusionsmodell rechnet in jedem einzelnen Schritt zwei Vorhersagen: eine mit deinem Prompt und eine ohne. Die Differenz zwischen beiden ist die Richtung, in die dein Prompt das Bild zieht; der Guidance-Wert ist der Faktor, mit dem diese Richtung verstärkt wird. Der Negativ-Prompt ersetzt nun einfach die zweite, eigentlich leere Vorhersage. Statt „von nichts weg, zu deinem Prompt hin“ rechnet das Modell dann „von deinem Negativ-Prompt weg, zu deinem Prompt hin“. Es ist derselbe Mechanismus, nur mit gefülltem Gegenpol.

Daraus folgen zwei Dinge, die man wissen sollte. Erstens: Der Negativ-Prompt kostet keine zusätzliche Rechenzeit, weil die zweite Vorhersage ohnehin berechnet wird. Zweitens: Er wirkt nur so stark, wie der Guidance-Wert es zulässt. Bei einem Guidance-Wert um 3 bis 4 bleibt der Negativ-Prompt ein sanfter Hinweis, bei 9 bis 11 wird er zur Vorschrift – mit allen Nebenwirkungen, die hohe Guidance sonst auch hat.

Wo der Negativ-Prompt ansetzt – und wo er ins Leere greift

Derselbe Wunsch, drei völlig verschiedene Maschinen darunter



Drei Maschinen, ein Wunsch: aktiv wegdrücken, ins Leere greifen oder auf einer ganz anderen Ebene ansetzen.

Eine Standardliste, die tatsächlich trägt

Es lohnt sich, eine kleine Grundliste zu führen, die du fast immer mitschickst, und sie pro Bildsorte um ein bis zwei Begriffe zu ergänzen. Wichtig ist das Wort „klein“. Die im Netz kursierenden Monsterlisten mit sechzig Begriffen sind keine Qualitäts-, sondern eine Stilvorgabe: Sie drücken das Bild so weit von so vielem weg, dass am Ende ein gleichförmiger Look übrig bleibt, den man in jedem Blogbeitrag wiedererkennt.

Problem	Negativ-Begriffe	Anmerkung
Schrift und Logos im Bild	text, watermark, signature, logo	der mit Abstand nützlichste Block
Zerfledderte Hände	extra fingers, fused fingers, deformed hands	hilft, löst das Problem aber nicht vollständig
Überzogene Farben	oversaturated, hdr, overexposed	oft Symptom eines zu hohen Guidance-Werts
Rahmen und Collagen	frame, border, split image, collage	besonders bei ungewöhnlichen Seitenverhältnissen
Ungewollte Zeichnungslook	cartoon, illustration, 3d render	nur bei fotorealistischem Ziel sinnvoll
Störende Menschen im Bild	people, crowd, person	bei destillierten Modellen wirkungslos

Auch im Negativ-Prompt kannst du gewichten. Ein hartnäckiger Störer lässt sich mit (watermark:1.3) spürbar stärker wegdrücken. Dieselben Grenzen gelten wie im positiven Prompt: Über 1,5 wird es unangenehm. Und wenn die Hände trotz allem nicht besser werden, liegt das an etwas Grundsätzlicherem – nachzulesen in [Warum KI keine Hände kann](#).

Wann der Negativ-Prompt schadet

Drei Fälle sehe ich immer wieder. Erstens: Begriffe im Negativ-Prompt, die auch im positiven stehen. Das ist kein cleverer Balanceakt, sondern eine Anweisung an das Modell, sich selbst zu widersprechen; das Ergebnis ist matschig. Zweitens: Negativ-Begriffe, die zu nah am gewünschten Motiv liegen. Wer „dark“ negativ setzt, verliert unter Umständen auch die Schatten, die das Bild

erst plastisch machen. Drittens: Listen, die über Jahre gewachsen sind und deren Herkunft niemand mehr kennt. Ein Negativ-Prompt aus der SD-1.5-Ära an einem heutigen Modell ist bestenfalls wirkungslos und schlimmstenfalls stilprägend.

Negativ-Prompt gehört in die Dokumentation, nicht in den Kopf

Der Negativ-Prompt ist Teil der Bildkonfiguration, genau wie Seed, Sampler und Schrittzahl. Wenn er nur in der Zwischenablage einer einzelnen Person existiert, ist deine Bildproduktion nicht reproduzierbar – und das fällt spätestens auf, wenn diese Person im Urlaub ist und eine Bildserie fortgesetzt werden muss.

Warum das Festhalten des Seeds dabei genauso wichtig ist, steht in [Der Seed bei der KI-Bildgenerierung](#).

Modelle ohne Negativ-Prompt: Flux und die destillierte Verwandtschaft

Und jetzt die Stelle, an der die meisten stolpern. Du wechselst auf ein Flux-Modell, pflegst deine bewährte Negativliste ein – und es passiert nichts. Kein Unterschied, kein Effekt, keine Fehlermeldung. Das Feld ist da, es nimmt Text entgegen, und der Text landet im Nirwana.

Warum FLUX.1 keinen Negativ-Prompt hat

Der Grund ist Guidance-Destillation. Black Forest Labs hat bei FLUX.1 [dev] und [schnell] das teure Zwei-Vorhersagen-Verfahren in das Modell selbst hineintrainiert: Das Modell hat gelernt, direkt das Ergebnis zu liefern, das ein klassisches Modell erst nach zwei Durchgängen und einer Extrapolation ausgespuckt hätte. Der gewünschte Guidance-Grad wird dabei als Zahl in das Modell hineingereicht – in ComfyUI ist das der FluxGuidance-Knoten, der auf der positiven Konditionierung sitzt, nicht im Sampler.

Damit läuft der eigentliche Sampler mit einem CFG-Wert von 1. Und bei CFG 1 wird die zweite, unkonditionierte Vorhersage gar nicht erst berechnet – ComfyUI überspringt sie als Geschwindigkeitsoptimierung. Es gibt schlicht keinen Gegen-Zweig, in den der Negativ-Prompt eingesetzt werden könnte. Das ist kein Versäumnis und kein Bug, sondern der Preis dafür, dass Flux bei gleicher Schrittzahl etwa doppelt so schnell rechnet wie ein klassisches Modell.

Dasselbe gilt für alle anderen destillierten Modelle: SD 3.5 Large Turbo, SDXL-Turbo, LCM- und Lightning-Varianten, FLUX.1 [schnell]. Wer wissen will, welcher Typ vor ihm liegt, schaut auf den empfohlenen CFG-Wert. Steht dort 1,0, brauchst du über das Negativ-Feld nicht weiter nachzudenken.

Die saubere Antwort: positiv umformulieren

Statt „keine Menschen“ in ein wirkungsloses Feld zu schreiben, sagst du im positiven Prompt, was stattdessen dort sein soll: „eine leere Werkhalle am frühen Morgen, die Maschinen stehen still, niemand ist im Bild“. Du beschreibst die gewünschte Abwesenheit als Anwesenheit von etwas anderem. Das klingt nach Wortklauberei und ist doch der entscheidende Unterschied: Ein Sprachmodell-Encoder versteht „niemand ist im Bild“ als Aussage über die Szene – während ein Stichwort „keine Menschen“ im CLIP-Zeitalter schlicht als „Menschen“ ankam, weil CLIP Verneinungen nicht kennt.

Dieselbe Logik funktioniert für alles andere. Statt „kein Text“ schreibst du „die Verpackung ist unbedruckt, glatte Oberfläche“. Statt „keine Unschärfe“ schreibst du „durchgehend scharf von vorn bis hinten, kleine Blende“. Statt „nicht überbelichtet“ schreibst du „gedämpftes Licht, weiche

Schatten, keine ausgefressenen Lichter“ – wobei hier der Nebensatz tatsächlich ankommt, weil der Encoder ihn als Beschreibung liest.

Die Umformulierung ist meistens die bessere Lösung

In der Praxis ist das positive Umschreiben nicht nur ein Notbehelf, sondern häufig das bessere Ergebnis. Ein Negativ-Prompt sagt dem Modell, wovon es weg soll – wohin es geht, bleibt offen. Eine positive Beschreibung gibt beides vor. Auch an Modellen mit funktionierendem Negativ-Feld lohnt es sich daher, erst den positiven Prompt zu präzisieren und die Negativliste klein zu halten.

Was heute wirklich funktioniert, wenn du trotzdem einen brauchst

Es gibt Fälle, in denen Umformulieren nicht reicht – hartnäckige Bildstörer, die sich sprachlich schlecht ausschließen lassen. Dafür haben sich inzwischen drei Verfahren durchgesetzt, die deutlich verlässlicher sind als die alten Bastellösungen.

Das erste ist NAG, Normalized Attention Guidance. Es setzt nicht an der Vorhersage an, sondern im Aufmerksamkeitsmechanismus des Modells: Der Negativ-Begriff wird dort normalisiert weggedreht, ganz ohne zweiten Rechenzweig. In ComfyUI ist NAG inzwischen als eingebauter Knoten NAGuidance verfügbar, also ohne Zusatzinstallation. Die Voreinstellungen sind `nag_scale` 5,0, `nag_alpha` 0,5 und `nag_tau` 1,5; für flussbasierte Modelle wie Flux hat sich ein `nag_sigma_end` um 0,75 bewährt, für SDXL eher ein Wert um 4.

Das zweite ist echtes CFG mit zweitem Durchgang. In der Diffusers-Bibliothek heißt der Parameter `true_cfg_scale`: Setzt du ihn über 1 und gibst einen Negativ-Prompt an, wird die unkonditionierte Vorhersage tatsächlich wieder berechnet, zusätzlich zur destillierten Guidance. Das funktioniert, kostet aber ungefähr die doppelte Rechenzeit und verträgt sich nicht mit jedem Modell gleich gut.

Das dritte ist VSF, Value Sign Flip, ein Verfahren speziell für Modelle mit sehr wenigen Schritten, bei denen NAG und klassisches CFG beide an ihre Grenzen stoßen. Es ist noch näher an der Forschung als an der Werkbank, taucht aber zunehmend in Werkzeugkästen auf.

Die alten Umgehungen – Dynamic Thresholding, das den CFG-Wert künstlich über 1 hebt und die Übersättigung nachträglich wieder wegrechnet – funktionieren weiterhin, sind aber nicht mehr die erste Wahl. Wer heute einen Negativ-Prompt an einem destillierten Modell braucht, greift zu NAG.

FLUX.2 verschiebt die Lage ein Stück

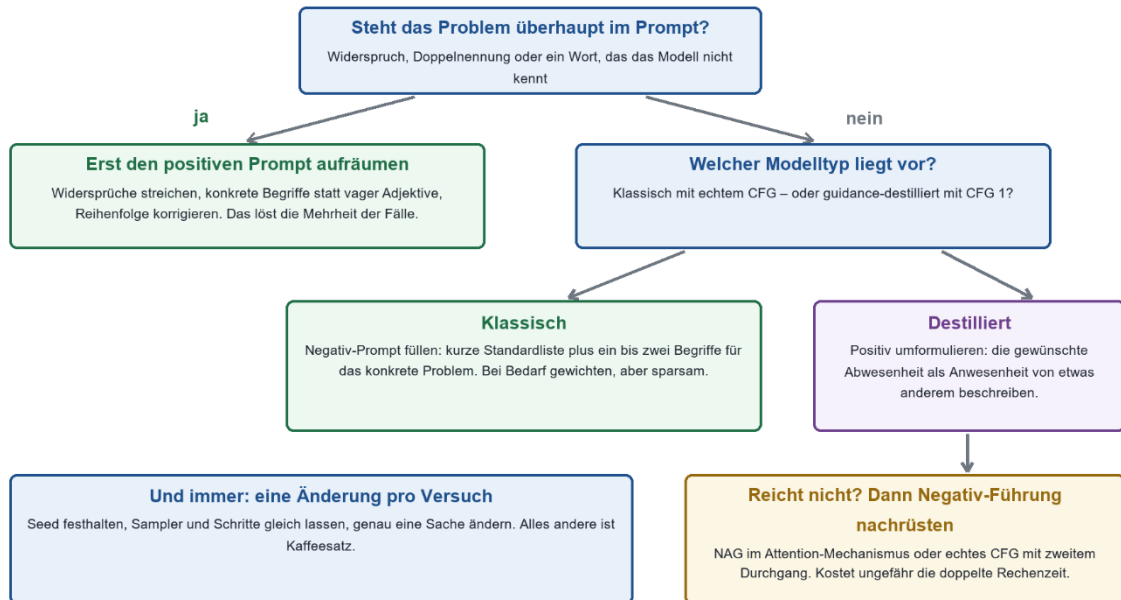
Mit FLUX.2 hat sich das Bild differenziert. FLUX.2 [dev] mit 32 Milliarden Parametern arbeitet weiterhin mit Guidance-Embeddings; der empfohlene Guidance-Wert liegt zwischen etwa 2,5 und 4, ein Negativ-Prompt im klassischen Sinn ist nicht vorgesehen. Die [klein]-Reihe dagegen wurde ohne Schritt- und Guidance-Destillation trainiert – dort ist echtes CFG mit Negativ-Prompt grundsätzlich möglich, auch wenn die meisten Beispiel-Abläufe weiterhin mit CFG 1,0 und leerem Negativ-Feld arbeiten. Die 4-Milliarden-Variante steht unter Apache 2.0 und läuft mit rund 13 GB Videospeicher, die 9-Milliarden-Variante braucht etwa 24 GB und fällt unter die nicht-kommerzielle FLUX-Lizenz. Das ist bei betrieblicher Nutzung kein Detail, sondern die erste Frage.

Modell	Guidance-Mechanik	Negativ-Prompt	Was du stattdessen tust
SD 1.5, SDXL	echtes CFG, 5 bis 9	wirkt stark	Standardliste pflegen
SD 3.5 Large	echtes CFG, 3,5 bis 5	wirkt	kurze Liste genügt
SD 3.5 Large Turbo	destilliert, CFG 1	wirkungslos	positiv umformulieren
FLUX.1 [dev]	Guidance-Embedding, CFG 1	wirkungslos	umformulieren oder NAG
FLUX.1 [schnell]	destilliert, CFG 1	wirkungslos	umformulieren oder NAG
FLUX.2 [dev]	Guidance-Embedding 2,5 bis 4	nicht vorgesehen	umformulieren

FLUX.2 [klein] Basis	kein Guidance-Distillat	möglich	CFG über 1 testen
Qwen-Image	CFG vorhanden	Feld ohne echte Wirkung	umformulieren

Unerwünschtes loswerden: der kurze Weg

Vier Fragen, danach weißt du, an welcher Schraube du drehst



Vier Fragen bis zur Entscheidung. Der häufigste Fall endet schon auf der zweiten Stufe: Der positive Prompt war einfach unsauber.

Vom Prompt-Zettel zum Verfahren

Alles bis hierhin war Handwerk für den Einzelfall. Im Unternehmen entscheidet aber etwas anderes über Erfolg oder Frust: ob das Wissen an einer Person hängt oder an einem Verfahren. Ein Prompt, der ein gutes Bild erzeugt hat, ist ein Betriebsmittel. Wenn er in einem Chatverlauf verschwindet, hast du das Betriebsmittel weggeworfen.

Prompts sind Konfiguration, keine Notizen

Ein vollständiger Bildeintrag besteht aus positivem Prompt, Negativ-Prompt, Modellname mit Version, Seed, Sampler, Schrittzahl, Guidance-Wert, Auflösung und gegebenenfalls den verwendeten LoRAs samt Stärke. Das sind neun Felder. Wer davon drei weglässt, kann ein Bild nicht rekonstruieren – und genau das wird irgendwann verlangt, sei es für eine Bildserie, für eine Nachbearbeitung oder für einen Nachweis.

Für den Teil, der sich um Werte, Katalog und Nachweis kümmert, setze ich UB.Dossier ein: Dort liegen die Prompt-Bausteine als gepflegter Katalog, nicht als gewachsene Textdatei, und jede Bildkonfiguration ist nachvollziehbar hinterlegt. Für den Produktionsteil – also den Ablauf selbst, seine Änderungen und den Plausibilitäts-Check vor dem Durchlauf – nutze ich UB.Forge. Wie das in der Praxis zusammenspielt, beschreibe ich in [Dogfooding: Ich setze ein, was ich berate](#). Wer die Ablage lieber auf vorhandener Microsoft-Infrastruktur aufbaut, findet den Weg dorthin in [Digital Asset Management für KI-Bilder mit SharePoint Online und Power Platform](#).

Prompt-Bausteine und Markenkonsistenz

Der praktische Durchbruch in Unternehmensprojekten kommt selten durch bessere Einzelprompts, sondern durch Bausteine. Du definierst einmal, wie „unser Produktfoto“ aussieht – Lichtführung, Perspektive, Hintergrund, Farbstimmung – und hältst das als festen Textblock fest. Variabel bleibt nur das Motiv. Dadurch sehen fünfzig Bilder aus fünf Abteilungen wie fünfzig Bilder derselben Marke aus und nicht wie eine Sammlung von Zufallsfunden.

Wie weit man diesen Gedanken treiben kann und wo die Grenze zwischen Prompt-Arbeit und Modelltraining verläuft, behandle ich in [Markenkonsistente KI-generierte Bilder](#). Die technische Verlängerung davon – ein eigens trainierter Zusatz für wiederkehrende Motive – steht in [Was ist ein LoRA?](#).

Dokumentation ist inzwischen auch Pflicht

Seit dem 2. August 2026 gelten die Transparenzpflichten des EU AI Acts in vollem Umfang. Für die Bildproduktion heißt das: KI-erzeugte Inhalte müssen maschinenlesbar gekennzeichnet sein, und wer Bilder veröffentlicht, muss auskunftsfähig sein. Eine saubere Prompt-Dokumentation ist damit vom Ordnungssinn zur Betriebsvoraussetzung geworden. Was konkret gilt, steht in [EU AI Act: Was seit dem 2. August 2026 gilt](#). Wie sich solche Bilder umgekehrt erkennen lassen, behandelt [KI-generierte Bilder erkennen](#).

Wenn du das Thema systematisch aufsetzen willst – vom Prompt-Katalog über die Modellauswahl bis zur Ablage – ist das der Kern meiner [KI-Bildgenerierung Beratung](#). Geht es eher darum, dass ein vorhandenes Team das Handwerk beherrscht und aufhört, Bilder zu würfeln, passt die [KI-Bildgenerierung Schulung](#) besser. Und wer Prompt-Texte selbst mit Sprachmodellen vorbereiten lässt, findet den passenden Rahmen unter [Claude im Unternehmen](#).

Was der Prompt nicht kann

Zum Schluss die unangenehme Wahrheit, die dir viel Zeit spart: Der beste Prompt zwingt ein Modell nicht zu etwas, das es nicht gelernt hat. Wenn ein Begriff im Training nicht vorkam oder nur mit zwei Dutzend Beispielen belegt ist, hilft keine noch so geschickte Formulierung. Ein Spezialwerkzeug aus deinem Maschinenbau, eine regionale Trachtenform, ein Firmenlogo – all das liegt außerhalb dessen, was Wortwahl lösen kann. Dann ist die Grenze das Modell, nicht deine Formulierung.

Genauso wenig löst der Prompt Widersprüche auf. Wer Frontalansicht und Blick über die Schulter gleichzeitig verlangt, bekommt einen Kompromiss, den niemand wollte. Wer sich fragt, warum eine völlig klare Anweisung trotzdem ignoriert wird, findet die systematischen Ursachen in [Warum die KI den Prompt ignoriert](#).

Und zuletzt: Gewichtung und Negativ-Prompt sind Feinwerkzeuge. Sie ersetzen weder einen durchdachten Grund-Prompt noch vernünftige Grundeinstellungen. Wenn das Bild grundsätzlich nicht funktioniert, liegt es häufiger an der Schrittzahl, am Sampler oder an einer krummen Auflösung als an einem fehlenden Wort. Die Zusammenhänge dazu stehen in [Sampling-Steps erklärt](#), [Sampler & Scheduler erklärt](#) und [Auflösung & Seitenverhältnis bei KI-Bildern](#).

Fazit

Ein Prompt in der KI-Bildgenerierung ist keine Beschwörungsformel, sondern eine Bauanleitung mit fünf Bausteinen: Bildart, Hauptmotiv, Merkmale, Umgebung, Stil und Licht – in dieser Reihenfolge. Ob du sie als Kommakette oder als ausformulierten Absatz schreibst, entscheidet nicht dein Geschmack, sondern der Text-Encoder deines Modells. CLIP-Modelle wollen Tags in 75-Token-Blöcken, die heutigen Modelle mit großen Sprachmodellen als Encoder wollen ganze Sätze und verarbeiten bis zu 512 Token.

Die Gewichtung in Klammern ist ein Feinwerkzeug mit einem schmalen Arbeitsbereich zwischen 1,1 und 1,4 – und mit der Tücke, dass dieselbe Zahl in verschiedenen Oberflächen unterschiedlich hart wirkt. Bei Modellen mit Sprachmodell-Encoder ersetzt du sie besser durch präzisere Sprache.

Der Negativ-Prompt ist bei klassischen Modellen mit echtem Guidance-Wert eines der wirksamsten Werkzeuge überhaupt, kostet nichts extra und sollte kurz bleiben. Bei guidance-destillierten Modellen – FLUX.1, FLUX.2 [dev], alle Turbo- und Schnell-Varianten – existiert das Feld, wirkt aber nicht, weil der Gegen-Zweig fehlt, auf dem der Mechanismus beruht. Die saubere Antwort dort ist positives Umformulieren: die gewünschte Abwesenheit als Anwesenheit von etwas anderem beschreiben. Reicht das nicht, gibt es mit NAG heute eine in ComfyUI eingebaute, verlässliche Negativ-Führung, die ohne CFG auskommt.

Und über allem steht die langweiligste Regel, die aber am meisten bringt: eine Änderung pro Versuch, Seed festhalten, Ergebnis notieren. Wer das drei Wochen durchhält, hat am Ende eine Prompt-Bibliothek statt einer Sammlung glücklicher Zufälle. Alles Weitere zum Gesamtbild findest du im Überblick [KI-Bildgenerierung](#).

Häufige Fragen zu Prompt und Negativ-Prompt

Wie baue ich einen guten Prompt auf?

In fünf Bausteinen, vom Wichtigsten zum Beiwerk: Bildart, Hauptmotiv, Merkmale des Motivs, Umgebung, zuletzt Licht und Stil. Die Bildart – Produktfoto, Illustration, technische Zeichnung – ist der am häufigsten vergessene und zugleich wirksamste Baustein. Konkrete Begriffe schlagen vage Adjektive: „weiches Fensterlicht von links“ bringt mehr als „schönes Licht“.

Stichwortliste oder ganzer Satz?

Das hängt am Text-Encoder deines Modells. SD 1.5, SDXL und deren Ableger arbeiten mit CLIP und bevorzugen Kommaketten aus Stichworten. FLUX.1, FLUX.2, Qwen-Image und Z-Image nutzen große Sprachmodelle als Encoder und liefern mit ausformulierten Sätzen deutlich bessere Ergebnisse – inklusive räumlicher Beziehungen und Nebensätzen. Bei FLUX.2 [klein] liegt der Arbeitsbereich etwa bei 40 bis 120 Wörtern.

Sind längere Prompts besser?

Nicht automatisch. Bei CLIP-Modellen wird ein langer Prompt in Blöcke von 75 Token zerlegt, und was hinten steht, geht leichter unter; zehn bis fünfzehn treffende Stichworte wirken dort besser als dreißig schwammige. Bei Modellen mit großem Encoder und 512 Token Kapazität lohnt sich Länge durchaus – aber nur, wenn jeder Satz eine zusätzliche Information trägt. Füllwörter kosten Aufmerksamkeit und liefern nichts.

Was bedeutet die Zahl in Klammern wie (rote Jacke:1.3)?

Das ist die Gewichtung. Der Wert 1.0 ist neutral, darüber wird der Baustein betont, darunter zurückgenommen. Der Arbeitsbereich liegt zwischen 1,1 und 1,4; ab etwa 1,8 wird das Bild steif, die Farben kippen und es tauchen Artefakte auf. Abschwächen mit Werten unter 1,0 funktioniert dabei verlässlicher als Verstärken.

Warum wirkt dieselbe Gewichtung in zwei Programmen unterschiedlich?

Weil die Oberflächen unterschiedlich rechnen. Werkzeuge der Automatic1111-Familie normalisieren die Prompt-Einbettung nach dem Gewichten wieder auf ihren ursprünglichen Mittelwert; ComfyUI multipliziert in der Voreinstellung direkt ohne Ausgleich. Dieselbe 1.3 wirkt

deshalb in ComfyUI spürbar härter. Übernommene Werte aus fremden Prompts solltest du dort ein bis zwei Zehntel niedriger ansetzen.

Wozu ist der Negativ-Prompt da?

In den Negativ-Prompt kommt, was nicht im Bild sein soll: Text, Wasserzeichen, verunglückte Hände, überzogene Farben, Rahmen. Das Modell ersetzt damit seine unkonditionierte Gegen-Vorhersage und schiebt das Ergebnis aktiv von diesen Begriffen weg. Zusätzliche Rechenzeit kostet das nicht, weil die zweite Vorhersage bei echtem CFG ohnehin berechnet wird.

Wie sieht eine brauchbare Standard-Negativliste aus?

Kurz. Vier bis acht Begriffe gegen die üblichen Störer reichen: text, watermark, signature, logo als Block gegen Schrift im Bild, dazu je nach Bildsorte etwas gegen Rahmen oder gegen zerfledderte Hände. Die kursierenden Sechzig-Begriffe-Listen sind keine Qualitätsvorgabe, sondern eine Stilvorgabe – sie drücken das Bild so weit von so vielem weg, dass ein uniformer Look übrig bleibt.

Warum hat Flux keinen Negativ-Prompt?

Weil FLUX.1 guidance-destilliert ist. Das Zwei-Vorhersagen-Verfahren wurde ins Modell hineintrainiert, der gewünschte Guidance-Grad wird als Zahl auf der positiven Konditionierung übergeben, und der Sampler läuft mit einem CFG-Wert von 1. Bei CFG 1 wird die unkonditionierte Vorhersage gar nicht erst berechnet – es fehlt also der Gegen-Zweig, in den ein Negativ-Prompt eingesetzt würde. Das Feld ist in vielen Oberflächen sichtbar, bewirkt aber nichts.

Wie werde ich bei Flux Unerwünschtes los?

Indem du positiv umformulierst. Statt „keine Menschen“ schreibst du, was stattdessen da sein soll: „eine leere Werkhalle am frühen Morgen, die Maschinen stehen still“. Statt „kein Text“ schreibst du „unbedruckte, glatte Verpackung“. Du beschreibst die gewünschte Abwesenheit als Anwesenheit von etwas anderem – das versteht der Encoder, eine Verbotsliste versteht er nicht.

Was ist NAG und wann lohnt es sich?

NAG steht für Normalized Attention Guidance und ist ein Verfahren, das Negativ-Begriffe im Aufmerksamkeitsmechanismus des Modells wegdreht statt über einen zweiten Rechenzweig. Damit funktioniert Negativ-Führung auch an destillierten Modellen mit CFG 1. In ComfyUI ist NAG als eingebauter Knoten NAGuidance verfügbar; die Voreinstellungen sind nag_scale 5,0, nag_alpha 0,5 und nag_tau 1,5. Es lohnt sich bei hartnäckigen Störern, die sich sprachlich schlecht ausschließen lassen – für den Alltag bleibt das positive Umformulieren der kürzere Weg.

Gilt das alles auch für FLUX.2?

Teilweise. FLUX.2 [dev] arbeitet weiterhin mit Guidance-Embeddings, empfohlen zwischen etwa 2,5 und 4, und kennt keinen klassischen Negativ-Prompt. Die [klein]-Reihe wurde dagegen ohne Schritt- und Guidance-Destillation trainiert, dort ist echtes CFG mit Negativ-Prompt grundsätzlich möglich. Beim Prompt-Stil gilt für alle FLUX.2-Varianten dasselbe: ausformulierte Beschreibung statt Tag-Kette.

Warum ignoriert das Modell einzelne Wörter komplett?

Drei übliche Gründe. Erstens: Der Begriff kam im Training kaum vor – dann hilft keine Formulierung, sondern nur ein anderes Modell oder ein trainierter Zusatz. Zweitens: Es gibt einen Widerspruch im Prompt, und das Modell liefert einen Kompromiss. Drittens: Der Begriff steht so weit hinten in einem langen Prompt, dass er gegen die vorderen Bausteine nicht ankommt. Vorziehen ist dann wirksamer als Gewichten.

Wie teste ich Prompt-Änderungen sauber?

Eine Änderung pro Versuch, Seed festhalten, Sampler, Schrittzahl und Guidance unverändert lassen. Nur dann weißt du, was die Änderung bewirkt hat. Wer gleichzeitig drei Dinge verstellt, produziert Bilder, aber keine Erkenntnis – und darf in zwei Wochen wieder von vorn anfangen.

Passend zum Thema

- › Wie stark das Modell dem Prompt insgesamt folgt: [CFG-Scale: Guidance in KI-Bildgeneratoren](#)
- › Warum eine klare Anweisung trotzdem ignoriert wird: [Warum die KI den Prompt ignoriert](#)
- › Warum du beim Testen den Seed festhalten solltest: [Der Seed bei der KI-Bildgenerierung](#)
- › Welcher Modelltyp überhaupt vor dir liegt: [Checkpoints, Modelle und VAE](#)
- › Wenn Prompt-Arbeit nicht mehr reicht: [Was ist ein LoRA?](#)
- › Grundlagen zum Verfahren dahinter: [Basiswissen KI-Bildgenerierung: Wie Diffusion funktioniert](#)
- › Prompt-Bibliothek in eine Produktion überführen: [Aufbau einer skalierbaren Bild-Maschine mit ComfyUI und Flux](#)
- › Wo die Bilder am Ende entstehen sollen: [Cloud vs. lokale KI-Bildgenerierung](#) · [KI-Bildgenerierung im Unternehmen – Die strategische Alternative zur Cloud](#)
- › Prompt-Arbeit direkt in Office: [Black Forest FLUX.2 in PowerPoint und Copilot](#)
- › Wie viele Varianten pro Durchlauf sinnvoll sind: [batch_size und Reproduzierbarkeit](#)