

Was ist ein LoRA?



Ein Schrank voller kleiner Dateien, ein großes Modell, das sich nicht verändert, und ein Regler, der über alles entscheidet. Bild selbst erzeugt und als KI-generiert gekennzeichnet.

Wissen	Beratung	Schulung
Alle Grundlagen zur KI-Bildgenerierung an einer Stelle – vom Diffusionsprinzip bis zur Governance.	Eigene LoRAs, markenkonsistente Bildsprache und ein Nachweis, der einer Prüfung standhält.	Dein Team lernt, wann ein LoRA das richtige Werkzeug ist – und wann der Prompt gereicht hätte.
<u>KI-Bildgenerierung</u>	<u>KI-Bildgenerierung Beratung</u>	<u>KI-Bildgenerierung Schulung</u>

Es gibt in jedem Bildprojekt diesen einen Nachmittag. Der Prompt ist zum zwanzigsten Mal umgebaut, die Negativliste sieht aus wie eine Beschwerde, der Sampler ist durchprobiert – und das Bild sieht immer noch aus wie von irgendwem, nicht wie von euch. Die Gesichter wechseln von Bild zu Bild, der Look ist beliebig, das Produkt stimmt im Detail nicht. An diesem Nachmittag lernen die meisten Teams den Satz, den es im Werkzeugkasten der Bildgenerierung gibt: Manche Probleme löst kein Prompt, dafür brauchst du ein LoRA.

Ich habe das bei einem Maschinenbauer erlebt, der eine Bildstrecke für den Messeauftritt brauchte. Die Marketingabteilung hatte drei Wochen gepromptet, das Ergebnis war technisch sauber und markenmäßig wertlos: jedes Bild ein anderer Farbraum, jedes Bild eine andere Lichtstimmung, keins davon erkennbar als die Firma. Wir haben dann an einem Tag ein einziges Stil-LoRA auf vierzig eigenen Fotografien trainiert. Danach sahen die Bilder aus wie aus einer Hand, und zwar aus der richtigen. Der Prompt war anschließend kürzer als vorher.

Das Gute an der Sache: Du musst dafür kein neues Riesenmodell herunterladen und auch keins von Grund auf trainieren. Ein LoRA ist eine vergleichsweise kleine Zusatzdatei, die du auf ein bestehendes Modell legst – und plötzlich kann das Modell etwas, das es vorher nicht konnte. In diesem Beitrag steht, was dabei technisch passiert, wie du die Stärke einstellst, welche Sorten es

gibt, warum Kompatibilität die häufigste Fehlerquelle ist und wie du zu einem eigenen LoRA kommst.

Was ein LoRA technisch ist

LoRA steht für Low-Rank Adaptation, sinngemäß: Anpassung über wenige Stellschrauben. Das Verfahren stammt nicht aus der Bildgenerierung, sondern wurde 2021 bei Microsoft Research für große Sprachmodelle entwickelt. Die Idee dahinter war schlicht ein Kostenproblem: Ein Modell mit 175 Milliarden Parametern komplett nachzutrainieren, kann sich kaum jemand leisten. Die Autoren zeigten, dass man die Zahl der trainierbaren Parameter um den Faktor 10.000 senken kann und trotzdem Ergebnisqualität auf dem Niveau des vollständigen Feintunings bekommt. Die Bildgenerierungsszene hat sich das kurz darauf ausgeliehen, und heute ist LoRA dort die mit Abstand verbreitetste Form der Anpassung. Wenn du irgendwo über LoRA Stable Diffusion stolperst, geht es genau um dieses Verfahren, angewandt auf ein Bildmodell.

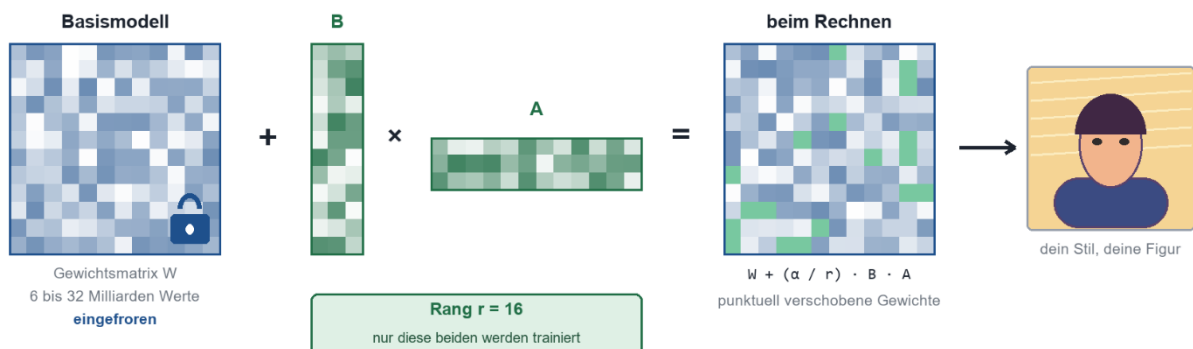
Das eingefrorene Modell und der schmale Aufsatz

Ein Bildmodell besteht im Kern aus sehr großen Zahlenmatrizen. Beim klassischen Feintuning werden diese Matrizen verändert – alle, überall, und das Ergebnis ist wieder ein vollständiges Modell in Originalgröße. Bei LoRA passiert etwas anderes: Die ursprünglichen Gewichte werden eingefroren und nicht angefasst. Stattdessen kommen pro Schicht zwei zusätzliche, sehr schmale Matrizen dazu, üblicherweise A und B genannt. Nur die werden trainiert. Beim Rechnen wird ihr Produkt auf die eingefrorenen Gewichte addiert.

Der Trick steckt in der Schmalheit. Wenn die Originalmatrix beispielsweise 3072 mal 3072 Werte hat, dann hat A die Form 16 mal 3072 und B die Form 3072 mal 16. Ihr Produkt hat wieder die volle Größe – aber gespeichert werden müssen nur die beiden schmalen Matrizen. Aus rund neun Millionen Werten werden knapp hunderttausend. Das ist der ganze Zauber, und deshalb heißt es Low Rank: Die Änderung wird durch eine Matrix von niedrigem Rang angenähert.

Low-Rank Adaptation: was beim Laden eines LoRA wirklich passiert

Das Basismodell bleibt unverändert. Trainiert wird nur der schmale Aufsatz.



Weil nur A und B gespeichert werden, wiegt die Datei einen Bruchteil des Modells – und das Training passt auf eine einzelne Grafikkarte.

Dasselbe Verfahren, das 2021 bei Microsoft Research für Sprachmodelle erfunden wurde, treibt heute jede Stil- und Figurenanpassung in der Bildgenerierung.

Skizze 1: Das Basismodell bleibt unverändert. Trainiert und gespeichert werden nur die beiden schmalen Matrizen, deren Produkt beim Rechnen aufaddiert wird.

Rang und Alpha: die zwei Zahlen im Dateinamen

Beim Herunterladen eines LoRA stolperst du über zwei Angaben, die zusammengehören. Der Rang – meist als rank, dim oder network_dim angegeben – ist die Breite der beiden Zusatzmatrizen. Er bestimmt, wie viel Information das LoRA überhaupt aufnehmen kann, und damit auch die

Dateigröße. Alpha ist ein Skalierungsfaktor: Der Beitrag des LoRA wird beim Rechnen mit Alpha geteilt durch Rang multipliziert. Sind beide gleich, ist der Faktor eins. Ist Alpha halb so groß wie der Rang, wirkt das LoRA von Haus aus nur halb so kräftig – was du dann beim Weight wieder ausgleichen musst.

Praktisch heißt das: Ein Rang von 4 bis 16 reicht für Stile völlig aus, für Gesichter und Objekte nimmt man eher 16 bis 32, und alles darüber ist meistens nur noch teurer, nicht besser. Die verbreiteten Trainingswerkzeuge fahren heute standardmäßig mit Rang 16 und Alpha 16, einer Lernrate von 0,0001, Stapelgröße 1 und rund 2.000 Trainingsschritten. Wer mit diesen Werten startet, macht selten etwas grundsätzlich falsch.

Nicht mehr „ein paar Megabyte“: die Dateigrößen von heute

Es hält sich hartnäckig die Vorstellung, LoRA-Dateien seien winzig. Das stimmte in der Ära von Stable Diffusion 1.5, wo ein Modell knapp zwei Gigabyte wog und ein LoRA tatsächlich unter hundert Megabyte blieb. Die Basismodelle sind seither erheblich gewachsen, und die LoRAs sind mitgewachsen – denn ihre Größe hängt unmittelbar an der Größe der Schichten, an die sie andocken. Klein sind sie trotzdem, aber eben nur im Verhältnis.

Basismodell	Größenordnung Modell	Parameter	LoRA bei Rang 16 bis 32
Stable Diffusion 1.5	ca. 2 GB	ca. 0,9 Mrd.	40 bis 150 MB
SDXL	ca. 6,5 GB	ca. 2,6 Mrd.	100 bis 400 MB
Stable Diffusion 3.5 Large	ca. 16 GB	ca. 8 Mrd.	150 bis 500 MB
FLUX.1 [dev]	ca. 24 GB	12 Mrd.	150 bis 400 MB
Qwen-Image	ca. 40 GB	20 Mrd.	300 bis 800 MB
FLUX.2 [dev]	ca. 64 GB	32 Mrd.	400 MB bis über 1 GB
Z-Image	ca. 12 GB	6 Mrd.	80 bis 250 MB

Die Zahlen sind Größenordnungen, keine Garantien: Wer bei Rang 64 trainiert und in voller Genauigkeit speichert, landet locker beim Dreifachen. Für die Praxis ist aber nur eine Botschaft wichtig – ein LoRA bleibt ein Bruchteil des Modells, und du kannst dutzende davon vorhalten, ohne dass die Festplatte in die Knie geht.

Was ein vollständiges Basismodell ausmacht und warum in einem Modellordner noch andere Dateien liegen, erklärt [Checkpoints, Modelle und VAE](#). Für die Einordnung, welche Modellfamilien heute überhaupt eine Rolle spielen, lohnt der Blick in [Die 15 wichtigsten Modelle zur lokalen Bildgenerierung](#).

Faktenkasten: LoRA in Zahlen

Erfunden 2021 bei Microsoft Research, ursprünglich für Sprachmodelle, nicht für Bilder.
 Reduktion der trainierbaren Parameter gegenüber vollem Feintuning: bis Faktor 10.000.
 Übliche Standardwerte heutiger Trainingswerkzeuge: Rang 16, Alpha 16, Lernrate 0,0001, 2.000 Schritte, Zwischenstände alle 250 Schritte, Ausgabe als safetensors in halber Genauigkeit.
 Trainingsdauer für ein Stil- oder Figuren-LoRA auf einer 24-GB-Karte: ein bis zwei Stunden.
 Speicherbedarf beim Training, je nach Familie: rund 16 GB für Z-Image, 24 GB für FLUX.1, Qwen-Image und FLUX.2 [dev], 32 GB für Qwen-Image-Edit, 48 GB für HiDream.
 Dateigröße eines typischen LoRA: zwischen einem halben und zwei Prozent des Basismodells.

Das Weight: der Regler, den du täglich anfasst

Jedes geladene LoRA bekommt eine Gewichtung mit – je nach Oberfläche heißt sie Weight, Strength, Multiplier oder schlicht Stärke. Sie legt fest, wie kräftig der aufaddierte Beitrag durchschlägt. Bei 0 ist das LoRA praktisch abgeschaltet, bei 1,0 wirkt es mit der Stärke, mit der es trainiert wurde. In den meisten Werkzeugen ist 1,0 die Voreinstellung, und der Eingabebereich reicht technisch weit darüber hinaus – in ComfyUI etwa von minus hundert bis plus hundert in Schritten von 0,01. Dass etwas eingebbar ist, heißt allerdings nicht, dass es sinnvoll ist.

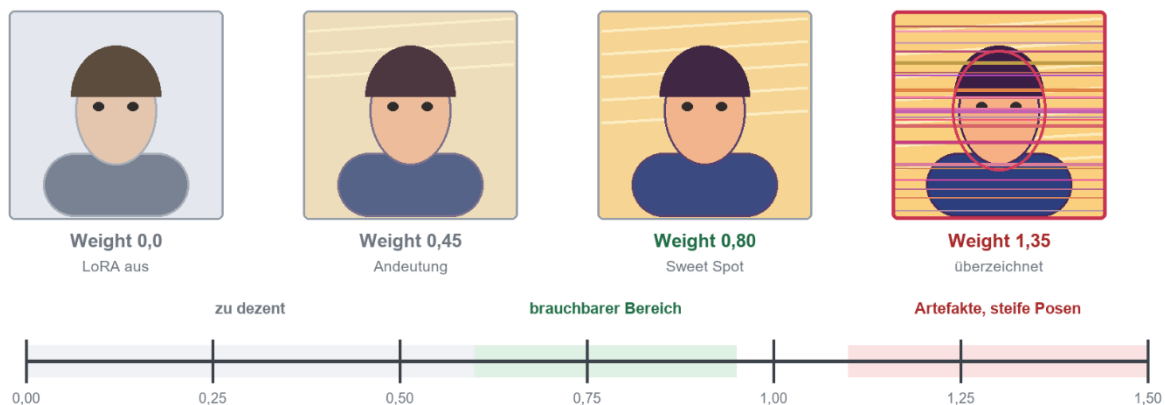
Wo der brauchbare Bereich liegt

Die naheliegende Annahme – voll aufdrehen für vollen Effekt – führt zuverlässig in die Irre. Ein zu hohes Weight überzeichnet: Gesichter werden wächsern, Posen steif, Farben knallen, und an den Kanten tauchen Doppelkonturen auf. Für die meisten Stil- und Figuren-LoRAs liegt der brauchbare Bereich zwischen 0,6 und 0,9. Das ist stark genug, dass der Effekt eindeutig zu sehen ist, und schwach genug, dass das Basismodell noch mitreden darf.

Ein wichtiger Zusatz, der oft als Regel verkauft wird und keine ist: Dieser Bereich hängt daran, wie stark das LoRA trainiert wurde. Ein knapp trainiertes LoRA braucht 1,0 und mehr, um überhaupt sichtbar zu werden. Ein übertrainiertes sitzt schon bei 0,4. Die Angabe auf der Downloadseite ist deshalb der bessere Startwert als jede Faustregel – und wenn dort nichts steht, fängst du bei 0,7 an und arbeitest dich in Schritten von 0,1 vor.

Das Weight: ein Regler, drei Zonen

Gleicher Prompt, gleicher Seed, gleiches LoRA – nur die Stärke ändert sich.



Der brauchbare Bereich ist kein Naturgesetz: Er hängt davon ab, wie stark das LoRA trainiert wurde. Manche sitzen bei 0,4, manche erst bei 1,0.

Deshalb testest du die Stärke immer bei festem Seed – sonst misst du den Zufall und nicht das LoRA.

Skizze 2: Dieselbe Szene bei vier Weights. Unterhalb von 0,6 passiert wenig, zwischen 0,6 und 0,9 liegt meist der brauchbare Bereich, oberhalb von 1,0 kippt das Bild.

Tipp: Stärke testen heißt Zufall ausschalten

Wenn du das Weight beurteilen willst, darf sich nichts anderes ändern. Halte den Seed fest, lass Prompt, Sampler, Schrittzahl und Auflösung unangetastet und variiere nur die eine Zahl. Sonst vermischt sich die Wirkung des LoRA mit dem Bildrauschen, und du schreibst einem LoRA eine Eigenschaft zu, die vom Zufallsstartwert kommt.

Wie der Seed funktioniert und warum er die Grundlage jeder belastbaren Testreihe ist, steht in [Der Seed bei der KI-Bildgenerierung](#). Wenn du mehrere Varianten gleichzeitig rechnen lässt, lohnt vorher ein Blick in [batch size und Reproduzierbarkeit](#).

Zwei Stärken statt einer

In ComfyUI und verwandten Oberflächen hat der Ladeknoten für ein LoRA nicht einen, sondern zwei Regler: einen für das Diffusionsmodell und einen für den Text-Encoder. Der erste bestimmt, wie stark das LoRA das Bild formt. Der zweite bestimmt, wie stark es die Bedeutung deiner Wörter verschiebt. Das ist nützlicher, als es klingt: Bei kräftigen Stil-LoRAs will man oft eine hohe Modellstärke, aber eine niedrige Text-Encoder-Stärke, weil sonst der Prompt an Präzision verliert und plötzlich Begriffe ignoriert werden.

Viele neuere LoRAs für FLUX und Qwen-Image enthalten gar keine Text-Encoder-Schichten mehr – dann bleibt der zweite Regler schlicht ohne Wirkung, egal was du einstellst. Das ist kein Fehler, sondern eine Folge davon, wie diese Modelle Sprache verarbeiten. Wenn dir der Prompt trotzdem entgleitet, hilft [Warum die KI den Prompt ignoriert](#) bei der Ursachensuche weiter.

Werte über 1 und negative Werte

Über 1,0 zu gehen ist gelegentlich vertretbar, nämlich bei LoRAs, die zu vorsichtig trainiert wurden. Bis etwa 1,3 kann das funktionieren. Darüber hinaus wird es selten schöner – man erkaufte sich mehr Effekt mit Artefakten, die nachher kein Nachbearbeitungsschritt mehr geraderückt.

Negative Werte sind der interessantere Sonderfall. Wenn du ein Detail-LoRA, das Schärfe hinzufügt, mit negativem Weight lädst, glättet es stattdessen. Damit hast du einen Regler in beide Richtungen, und für weiche Hauttöne oder eine bewusst reduzierte Bildsprache ist das gelegentlich genau das Richtige. Verlass dich allerdings nicht darauf, dass jede Oberfläche das mitmacht: Manche Implementierung behandelt alles unterhalb von null einfach als aus. Einmal ausprobieren, dann weißt du es.

Warnung: mehrere LoRAs summieren sich

Du kannst mehrere LoRAs gleichzeitig laden, und das ist auch der Normalfall – ein Stil, eine Figur, vielleicht noch ein Detail-LoRA. Aber die Beiträge addieren sich auf dieselben Gewichte. Zwei LoRAs mit je 1,0 ergeben nicht doppelt so gute Bilder, sondern einen Brei, in dem beide Effekte gegeneinander arbeiten.

Praxiswert: Bei zwei LoRAs jeweils 0,6 bis 0,7, bei drei eher 0,5 bis 0,6. Wenn du über vier kommst, hast du in aller Regel ein Prompt-Problem und kein LoRA-Problem. Und wenn ein Stapel partout nicht harmoniert, nimm eins heraus statt eins hinzu.

Welche Arten von LoRAs es gibt

LoRAs werden für sehr unterschiedliche Zwecke trainiert, und die Sorte entscheidet über fast alles: über die Größe des Datensatzes, über den Rang, über den Weight, bei dem du landest, und darüber, ob du das Ding überhaupt selbst bauen willst.

Stil-LoRAs

Das ist die häufigste Sorte und die, nach der die meisten suchen. Ein Stil-LoRA bringt einen Look mit: eine Maltechnik, eine Comic-Ästhetik, die Anmutung einer bestimmten Epoche, die Farbwelt und Lichtführung einer Marke. Es interessiert sich nicht dafür, was auf dem Bild zu sehen ist, sondern nur dafür, wie es aussieht. Stil-LoRAs kommen mit wenig Rang aus, brauchen vergleichsweise kleine Datensätze und vertragen sich untereinander am schlechtesten – zwei Stile gleichzeitig ergeben meistens einen dritten, den niemand wollte.

Charakter- und Personen-LoRAs

Trainiert auf ein bestimmtes Gesicht, damit dieselbe Person über viele Bilder hinweg wiedererkennbar bleibt. Das ist die technisch anspruchsvollste Sorte, weil Menschen Gesichter mit

brutaler Genauigkeit lesen: Zwei Millimeter Abweichung am Nasenrücken, und die Figur ist eine andere. Entsprechend braucht es einen besseren Datensatz – verschiedene Blickwinkel, verschiedene Lichtsituationen, verschiedene Abstände – und einen höheren Rang. Wer eine reale Person abbilden will, hat zudem eine Einwilligungsfraße zu klären, und zwar bevor trainiert wird, nicht danach.

Konzept-, Produkt- und Objekt-LoRAs

Auf einen konkreten Gegenstand oder einen abstrakten Bildaufbau trainiert: ein bestimmtes Produkt, ein Logo, eine wiederkehrende Verpackung, aber auch Dinge wie eine bestimmte Kameraperspektive oder ein Kompositionsschema. Im kommerziellen Einsatz ist das die interessanteste Sorte, weil hier der harte Nutzen liegt: Wenn eure Maschine, euer Möbelstück oder eure Flasche in jedem generierten Bild korrekt aussieht, sparst du dir die halbe Retusche.

Detail- und Korrektur-LoRAs

Kein eigenes Motiv, sondern eine Verbesserung: mehr Mikrokontrast, plausiblere Haut, sauberere Hände, weiches Licht. Diese LoRAs laufen meist mit niedrigem Weight im Hintergrund mit und mildern typische Modellschwächen ab. Sie sind ein Pflaster, kein Heilmittel – warum bestimmte Bildteile so hartnäckig schwierig bleiben, erklärt [Warum KI keine Hände kann](#).

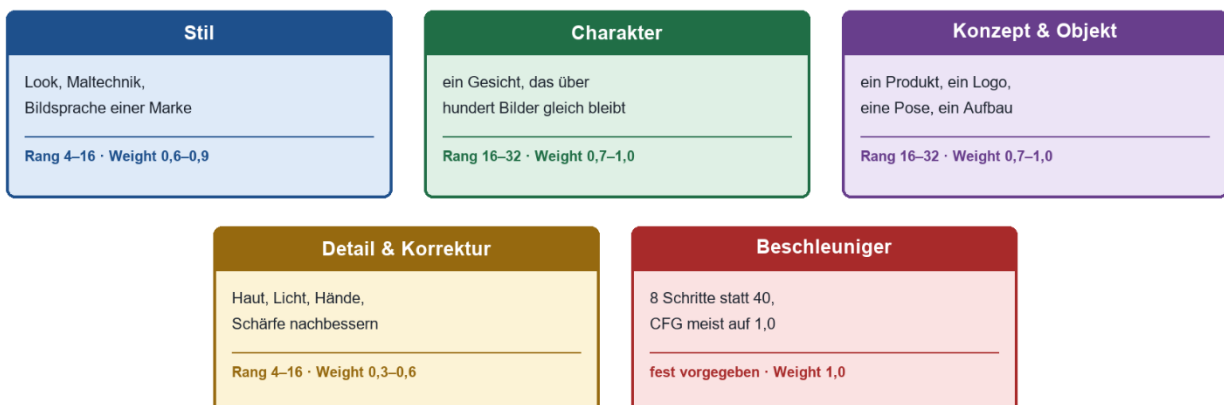
Beschleuniger-LoRAs

Die Sorte, die es in dieser Form vor wenigen Jahren noch nicht gab und die heute in fast jedem produktiven Aufbau steckt. Beschleuniger-LoRAs – bekannt unter Namen wie Lightning, Turbo oder Hyper – verändern nicht den Stil, sondern die Ökonomie: Sie destillieren das Modell so, dass es statt zwanzig bis fünfzig Schritten mit vier bis acht auskommt. Qwen-Image etwa rechnet normalerweise mit 20 bis 50 Schritten und mit dem passenden Lightning-LoRA mit 4 bis 8. Das ist ein Faktor fünf beim Durchsatz.

Der Haken: Diese LoRAs schreiben dir die anderen Regler vor. Die Schrittzahl ist festgelegt, die Guidance muss meist auf 1,0 stehen, und das Weight gehört auf den vorgesehenen Wert und nicht auf deinen Geschmack. Wie Schrittzahl und Guidance zusammenspielen, steht in [Sampling-Steps erklärt](#) und [CFG-Scale: Guidance in KI-Bildgeneratoren](#); welcher Sampler dazu passt, klärt [Sampler & Scheduler erklärt](#).

Fünf Sorten LoRA, fünf verschiedene Aufgaben

Die Sorte entscheidet über Datensatz, Rang und die Stärke, bei der du landest.



Stapeln geht – aber die Wirkung addiert sich. Drei LoRAs mit je 0,6 vertragen sich besser als zwei mit je 1,0.

Beschleuniger-LoRAs sind der Sonderfall: Die gehören auf ihren vorgesehenen Wert, nicht auf deinen Geschmack.

Ein Korrektur-LoRA repariert Details. Es macht aus einem schwachen Basismodell kein starkes.

Skizze 3: Fünf Sorten mit ihren typischen Rängen und Stärkebereichen. Die Beschleuniger-Sorte folgt eigenen Regeln.

Sorte	Wofür	Datensatz	Rang	Weight in der Praxis	Selbst bauen?
Stil	Look, Farbwelt, Bildsprache	15 bis 40 Bilder	4 bis 16	0,6 bis 0,9	ja, gut machbar
Charakter	wiedererkennbares Gesicht	25 bis 50 Bilder	16 bis 32	0,7 bis 1,0	ja, mit Sorgfalt
Konzept & Objekt	Produkt, Logo, Aufbau	20 bis 60 Bilder	16 bis 32	0,7 bis 1,0	ja, lohnt sich am meisten
Detail & Korrektur	Haut, Licht, Schärfe	sehr groß	4 bis 16	0,3 bis 0,6	eher fertig nehmen
Beschleuniger	weniger Schritte	Destillation	vorgegeben	1,0, wie dokumentiert	nein

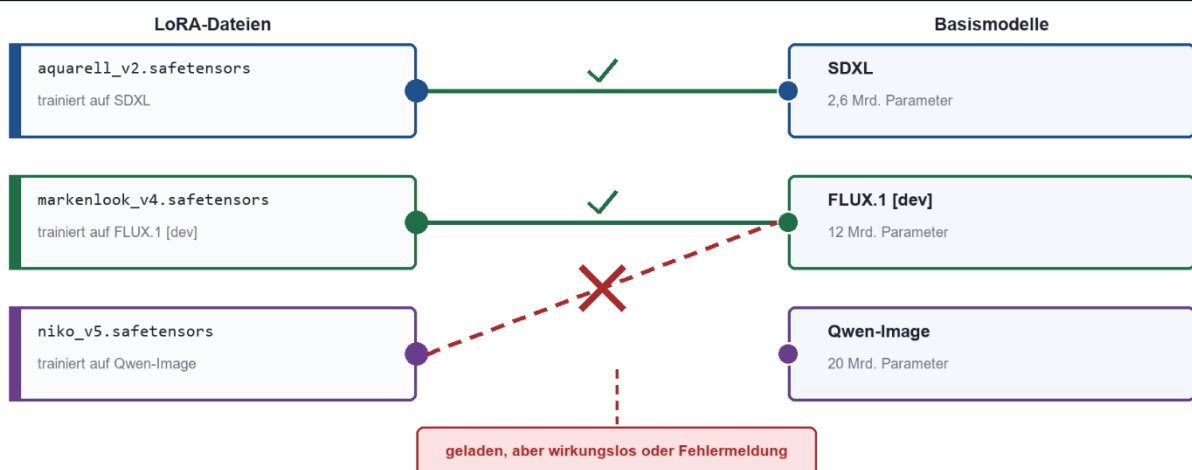
Kompatibilität: die häufigste Fehlerquelle überhaupt

Ein LoRA ist kein eigenständiges Programm, sondern eine Sammlung von Korrekturen für bestimmte Schichten eines bestimmten Modells. Es kennt die Namen dieser Schichten und ihre Abmessungen. Passt das nicht zusammen, passiert entweder gar nichts oder etwas Falsches. Deshalb gilt ohne Ausnahme: Ein LoRA funktioniert nur mit der Modellfamilie, für die es trainiert wurde.

Und „Familie“ ist enger gefasst, als der Name vermuten lässt. Ein LoRA für Stable Diffusion 1.5 läuft nicht auf SDXL. Ein LoRA für SDXL läuft nicht auf Stable Diffusion 3.5. Ein LoRA für FLUX.1 [dev] läuft nicht auf FLUX.2 [dev] – die Architektur ist von 12 auf 32 Milliarden Parameter gewachsen und dabei umgebaut worden. Innerhalb einer Familie funktioniert es dagegen meistens auch über verschiedene Feintunes hinweg: Ein SDXL-LoRA läuft in aller Regel auf jedem SDXL-Abkömmling, nur eben unterschiedlich gut, weil der Feintune den Ausgangspunkt schon verschoben hat.

Ein LoRA passt nur auf die Familie, für die es trainiert wurde

Nicht der Dateiname entscheidet, sondern die Architektur darunter.



Auch innerhalb einer Marke gilt keine Garantie: Ein LoRA für FLUX.1 läuft nicht auf FLUX.2, ein SD-1.5-LoRA nicht auf SDXL.

Manche Oberflächen laden es trotzdem klaglos. Dann siehst du keinen Fehler, sondern nur ein Bild, das seltsam aussieht.

Erste Frage bei jedem Download lautet deshalb: für welche Familie?

Skizze 4: Passende Verbindungen sind unauffällig. Die unpassende führt je nach Oberfläche zu einer Fehlermeldung – oder zu einem Bild, an dem du nur merkst, dass irgendetwas nicht stimmt.

Warnung: „Es lädt“ ist kein Kompatibilitätsnachweis

Manche Oberflächen überspringen unbekannte Schichten stillschweigend. Das LoRA erscheint dann in der Liste, das Bild wird gerechnet, es gibt keine Fehlermeldung – und die Wirkung ist teilweise oder gar nicht da. Das ist der unangenehmere Fall, weil du stundenlang am Prompt drehst, während das Problem eine Dateiendung weiter liegt.

Verlässlicher Gegentest: einmal mit Weight 0 und einmal mit Weight 1,0 rechnen, bei identischem Seed. Sind die beiden Bilder pixelgleich, tut das LoRA nichts. Dann ist entweder das Auslösewort nicht im Prompt oder das Basismodell das falsche.

Das Auslösewort

Zweithäufigste Ursache für ein scheinbar wirkungsloses LoRA: Viele LoRAs sind auf ein Auslösewort trainiert – ein künstliches Kürzel, das im Datensatz jede Bildunterschrift eingeleitet hat. Steht dieses Wort nicht im Prompt, wird das LoRA zwar geladen, findet aber keinen Anlass, sich einzumischen. Welches Wort das ist, steht auf der Downloadseite; bei selbst trainierten LoRAs hast du es selbst festgelegt. Was sonst noch in Prompt und Negativ-Prompt gehört, steht in [Prompt und Negativ-Prompt in der KI-Bildgenerierung](#).

Die Familien im Überblick

Für die Praxis relevant ist neben der Kompatibilität vor allem die Lizenzlage des Basismodells – denn die vererbt sich faktisch auf alles, was du damit produzierst. Der Stand im September 2026 sieht so aus:

Familie	Größe	Lizenz des Basismodells	LoRA-Angebot	Training möglich
SD 1.5	0,9 Mrd.	CreativeML Open RAIL-M	riesig, aber alternd	ja, sehr günstig
SDXL	2,6 Mrd.	CreativeML Open RAIL++-M	die mit Abstand größte Auswahl	ja, ab ca. 12 GB
SD 3.5 Large	8 Mrd.	Stability Community License	mittel	ja, ab 24 GB
FLUX.1 [dev]	12 Mrd.	FLUX [dev] Non-Commercial License	groß und gepflegt	ja, 24 GB
FLUX.2 [dev]	32 Mrd.	FLUX [dev] Non-Commercial License v2.0	im Aufbau	ja, 24 GB knapp, 32 GB komfortabel
FLUX.2 [klein]	4 Mrd.	Apache 2.0	im Aufbau	ja, moderat
Qwen-Image	20 Mrd.	Apache 2.0	wächst schnell	ja, 24 GB mit Quantisierung
Z-Image	6 Mrd.	Apache 2.0	jung, noch dünn	ja, ca. 16 GB

Zwei Punkte aus dieser Tabelle verdienen Aufmerksamkeit. Erstens: Die Stability-Community-Lizenz für Stable Diffusion 3.5 erlaubt kommerzielle Nutzung kostenfrei, aber nur bis zu einem Jahresumsatz von einer Million US-Dollar. Wer darüber liegt, braucht eine Enterprise-Lizenz – und zwar unabhängig davon, wie wenig KI in seinem Umsatz steckt. Zweitens: Die offenen Gewichte von FLUX.2 [dev] stehen unter einer ausdrücklich nicht-kommerziellen Lizenz. Das schöne LoRA, das du darauf trainiert hast, ist damit für Kundenarbeit nicht brauchbar, solange du nicht auf die kommerziell lizenzierte Variante oder auf ein Apache-2.0-Modell wie Qwen-Image, Z-Image oder FLUX.2 [klein] wechselst.

Tipp: Lizenz vor Look entscheiden

Die Reihenfolge, in der die meisten Teams vorgehen, ist verkehrt herum: erst ein Modell suchen, das schön aussieht, dann LoRAs dafür trainieren, dann die Lizenz lesen. Mach es andersherum. Kläre zuerst, was ihr rechtlich einsetzen dürft, und wählt die Familie danach aus. Ein LoRA ist an seine

Familie gebunden – ein Wechsel bedeutet, noch einmal von vorn zu trainieren.

Wie die Entscheidung zwischen eigener Hardware und Cloud-Diensten dabei hineinspielt, behandelt [Cloud vs. lokale KI-Bildgenerierung](#); die strategische Sicht auf den ganzen Aufbau steht in [KI-Bildgenerierung im Unternehmen – Die strategische Alternative zur Cloud](#).

Eigene LoRAs: der einzige verlässliche Weg zu markenkonsistenten Bildern

Bis hierher ging es um LoRAs, die jemand anderes gebaut hat. Der eigentliche Hebel im Unternehmen liegt woanders: bei den LoRAs, die du selbst trainierst. Denn Markenkonsistenz ist kein Prompt-Problem. Du kannst deine Hausfarben, deine Lichtführung und deine Bildsprache noch so präzise beschreiben – ein Modell, das auf Millionen fremder Bilder trainiert wurde, zieht mit jedem Bild ein Stückchen zurück in seinen Durchschnittsgeschmack. Ein LoRA verschiebt diesen Durchschnitt.

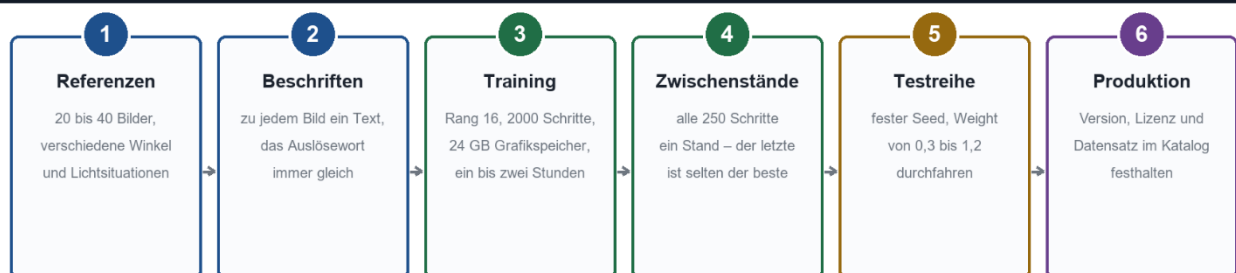
Was du dafür brauchst

Weniger, als die meisten befürchten. Ein Datensatz von zwanzig bis vierzig eigenen Bildern reicht für einen Stil, für ein Gesicht darf es etwas mehr sein. Dazu gehört zu jedem Bild eine Beschreibung als Textdatei – die Werkzeuge erwarten JPEG oder PNG mit gleichnamiger Textdatei daneben und übernehmen Zuschchnitt und Skalierung selbst. Und eine Grafikkarte mit 24 GB Speicher, was heute eine Consumer-Karte ist und keine Rechenzentrumshardware. Das Training selbst dauert ein bis zwei Stunden.

Der Aufwand liegt nicht beim Training, sondern beim Datensatz. Vierzig Bilder in gleicher Lichtsituation, aus gleichem Winkel, mit gleichem Hintergrund erzeugen ein LoRA, das genau diese eine Situation kann und sonst nichts. Variation im Datensatz ist das, was ein LoRA brauchbar macht. Und die Bildunterschriften müssen beschreiben, was variiert – nicht, was immer gleich ist. Was du in jeder Unterschrift wiederholst, lernt das LoRA als Konstante, und dann bekommst du sie nicht mehr los.

Vom Referenzbild zum eigenen LoRA

Sechs Stationen – die aufwendigste ist nicht das Training.



Der Aufwand liegt in Station 1 und 2. Wer den Datensatz schludert, trainiert Unschärfe mit ein.

Rechtlich zählt, was in Station 1 hineingeht: eigene Aufnahmen, geklärte Rechte, dokumentierte Herkunft.

Ein Datensatz aus fremden Bildern macht das Ergebnis nicht besser, sondern nur angreifbar.

Und ohne Station 6 weißt du in sechs Monaten nicht mehr, welche Version das gute Bild erzeugt hat.

Skizze 5: Der Ablauf vom Referenzbild bis zum produktiven Einsatz. Stationen 1 und 2 kosten die meiste Zeit und entscheiden über die Qualität.

Ein Beispiel aus dem eigenen Haus

Ich rede darüber nicht theoretisch. Niko, die Figur, die auf den Beitragsbildern dieser Website durch den Beratungsalltag stolpert, ist genau so entstanden: ein Satz eigener Referenzbilder, sorgfältig beschriftet, ein Charakter-LoRA darauf trainiert, eine Testreihe bei festem Seed über den ganzen Stärkebereich. Ohne das LoRA wäre Niko auf jedem Bild ein anderer Mensch, und die Website sähe aus wie eine Bilderdatenbank. Warum ich meine eigenen Werkzeuge im eigenen Betrieb einsetze, bevor ich sie empfehle, steht in [Dogfooding: Ich setze ein, was ich berate](#). Wie aus einzelnen guten Bildern eine konsistente Bildsprache wird, beschreibt [Markenkonsistente KI-generierte Bilder](#).

Was ich dabei gelernt habe und was in keinem Tutorial steht: Der letzte Trainingsstand ist fast nie der beste. Die Werkzeuge schreiben alle 250 Schritte einen Zwischenstand weg, und typischerweise sitzt das Optimum irgendwo in der Mitte. Danach wird die Ähnlichkeit größer und die Flexibilität kleiner – die Figur stimmt perfekt, lässt sich aber nicht mehr in eine andere Pose oder ein anderes Licht bringen. Man nennt das Übertraining, und man erkennt es nur, wenn man die Zwischenstände wirklich gegeneinander testet.

Ordnung halten, bevor es unübersichtlich wird

Nach einem halben Jahr produktiver Arbeit liegen in einem typischen Team zwischen zwanzig und achtzig LoRA-Dateien herum. Niemand weiß mehr, welche Version auf welchem Datensatz trainiert wurde, welche Lizenz daran hängt und welches Bild im Katalog aus welcher Kombination entstanden ist. Das ist kein hypothetisches Problem – das ist der Regelfall, und er wird genau dann teuer, wenn jemand nach der Herkunft eines veröffentlichten Bildes fragt.

Dagegen hilft nur, dass die Angaben an der Datei kleben und nicht im Kopf von irgendwem: Modellfamilie, Rang, Datensatzquelle, Lizenz des Basismodells, Auslösewort, getesteter Stärkebereich. In meinem eigenen Betrieb erledigt das UB.Dossier – dort liegt der Katalog, dort werden Werte geändert, dort steht der Nachweis, aus welchem Modell mit welchem LoRA ein bestimmtes Bild stammt. Die Produktion selbst läuft über UB.Forge: Dort wird der Workflow geändert, dort prüft ein Plausibilitäts-Check vor dem Lauf, ob LoRA und Basismodell zusammenpassen und ob die Stärke im dokumentierten Bereich liegt. Man kann das auch mit einer Tabelle und Disziplin lösen. Man muss es nur lösen.

Wie die Ablage und die Metadaten dazu sauber aufgesetzt werden, beschreibt [Digital Asset Management für KI-Bilder mit SharePoint Online und Power Platform](#); den technischen Unterbau für eine Produktion in Serie behandelt [Aufbau einer skalierbaren Bild-Maschine mit ComfyUI und Flux](#).

Wichtig: Kennzeichnung und Nachweis

Seit dem 2. August 2026 gelten die Transparenzpflichten des EU AI Acts. Wer synthetische Bilder veröffentlicht, muss sie als solche erkennbar machen. Ein eigenes LoRA ändert daran nichts – im Gegenteil: Je markenkonsistenter und realistischer eure Bilder werden, desto weniger fällt Betrachtern von selbst auf, dass sie KI-generiert sind.

Was konkret gefordert ist und wen es trifft, steht in [EU AI Act: Was seit dem 2. August 2026 gilt](#); woran sich generierte Bilder überhaupt erkennen lassen, behandelt [KI-generierte Bilder erkennen](#).

Was ein LoRA nicht kann

Ein LoRA ist ein Akzent, kein Umbau. Es verschiebt die Gewichte eines Modells ein Stück weit in eine Richtung – es gibt dem Modell keine Fähigkeit, die es nicht in Ansätzen schon hat. Wenn ein Basismodell mit einer Bildgattung grundsätzlich überfordert ist, reißt das kein LoRA heraus. Da hilft nur ein anderes Modell.

Ein LoRA ersetzt auch keinen guten Prompt. Es lenkt das Modell in eine Richtung, aber was auf dem Bild zu sehen ist, beschreibst weiterhin du. Ich habe Teams erlebt, die nach dem ersten erfolgreichen Stil-LoRA anfangen, den Prompt zusammenzustreichen – und sich wunderten, dass die Bilder beliebiger wurden. Ein LoRA macht den Prompt kürzer, weil du den Stil nicht mehr beschreiben musst. Es macht ihn nicht überflüssig.

Und ein LoRA ist keine Qualitätsgarantie. Schlecht trainierte LoRAs bringen Artefakte mit, die vorher nicht da waren, und bei hohem Weight bekommst du diese Artefakte in jedes Bild. Im Zweifel gilt, was fast immer gilt: weniger ist mehr. Wenn du unsicher bist, ob das Problem am LoRA, am Modell oder an den Einstellungen liegt, arbeite dich systematisch durch – die Grundlagen dazu stehen in [Basiswissen KI-Bildgenerierung: Wie Diffusion funktioniert](#). Auch die Bildgröße redet mit: Ein LoRA, das auf 1024 Pixel trainiert wurde, wird bei ganz anderen Formaten unzuverlässig, siehe [Auflösung & Seitenverhältnis bei KI-Bildern](#).

Lizenz und Recht: der Teil, den niemand gern liest

Im Hobbybereich fällt das kaum auf, im geschäftlichen Einsatz ist es der Punkt, an dem Projekte kippen. LoRAs werden von sehr unterschiedlichen Leuten erstellt und unter sehr unterschiedlichen Bedingungen veröffentlicht. Auf den einschlägigen Plattformen setzt jeder Anbieter seine Erlaubnisse selbst – manche erlauben kommerzielle Nutzung ohne Einschränkung, manche nur mit Namensnennung, manche verbieten sogar das Erzeugen von Bildern für Dritte. Eine plattformweite Regel gibt es nicht, du musst pro Datei nachsehen.

Dazu kommt die Frage, worauf das LoRA überhaupt trainiert wurde, und die ist unabhängig von der Lizenz. Ein LoRA auf ein geschütztes Markenprodukt oder auf eine reale Person ist heikel, auch wenn der Anbieter großzügig „kommerzielle Nutzung erlaubt“ angehängt hat – er kann Rechte, die ihm nicht gehören, nicht weitergeben. Bei selbst trainierten LoRAs hast du diese Frage selbst in der Hand, und das ist ein zusätzliches Argument für den eigenen Datensatz.

Warnung: drei Lizenzen, nicht eine

Bei jedem kommerziell genutzten Bild hängen mindestens drei Bedingungen übereinander: die des Basismodells, die des LoRA und die Rechtelage des Materials, auf dem das LoRA trainiert wurde. Es reicht nicht, eine davon zu prüfen – die restriktivste gewinnt.

Besonders unangenehm ist der Fall, in dem alles erlaubt aussieht: Basismodell frei, LoRA als kommerziell nutzbar markiert – und der Datensatz bestand aus Bildern eines fremden Fotografen.

Das ist keine Rechtsberatung, sondern der Hinweis, an welcher Stelle du genauer hinschauen solltest, bevor jemand anderes es tut.

Häufige Fragen zu LoRAs

Was ist ein LoRA einfach erklärt?

Ein LoRA ist eine kleine Zusatzdatei, die einem fertigen Bildmodell etwas Neues beibringt – einen Stil, ein Gesicht, ein Produkt, eine Detailverbesserung –, ohne das Modell selbst zu ersetzen. Das Basismodell bleibt unverändert; das LoRA legt sich beim Rechnen darüber und verschiebt die Gewichte ein Stück in die gewünschte Richtung. Deshalb kannst du LoRAs beliebig austauschen und kombinieren.

Was bedeutet das LoRA-Weight oder die LoRA-Stärke?

Das Weight legt fest, wie kräftig das LoRA wirkt. Bei 0 ist es aus, bei 1,0 wirkt es so stark, wie es trainiert wurde. Ein guter Startwert ist 0,7; der brauchbare Bereich liegt bei den meisten LoRAs

zwischen 0,6 und 0,9. Oberhalb von 1,0 wird das Bild schnell steif und artefaktreich. Steht auf der Downloadseite ein empfohlener Wert, ist der immer besser als jede Faustregel.

Warum tut mein LoRA scheinbar nichts?

Drei Ursachen decken fast alle Fälle ab. Erstens: Das Auslösewort fehlt im Prompt – viele LoRAs springen nur über ein bestimmtes Schlüsselwort an. Zweitens: Das Basismodell gehört zu einer anderen Familie, und die unpassenden Schichten werden stillschweigend übersprungen. Drittens: Das Weight steht auf einem Wert, bei dem dieses konkrete LoRA noch nichts tut. Der schnelle Test ist ein Vergleich bei identischem Seed mit Weight 0 gegen Weight 1,0 – sind die Bilder identisch, wirkt gar nichts.

Kann ich mehrere LoRAs gleichzeitig benutzen?

Ja, und das ist der Normalfall. Beachte aber, dass sich die Wirkungen auf denselben Gewichten addieren. Zwei LoRAs mit je 0,6 bis 0,7 vertragen sich deutlich besser als zwei mit je 1,0. Ab drei gehst du eher auf 0,5 bis 0,6 herunter. Zwei Stil-LoRAs gleichzeitig sind fast immer eine schlechte Idee, weil sie um dieselbe Wirkung konkurrieren – ein Stil plus eine Figur plus ein Detail-LoRA dagegen funktioniert gut.

Worin unterscheidet sich ein LoRA von einem Checkpoint?

Ein Checkpoint ist ein vollständiges Basismodell mit allen Gewichten und wiegt je nach Familie zwischen zwei und über sechzig Gigabyte. Ein LoRA ist nur die Differenz zu diesem Modell, in komprimierter Form, und ist ohne passenden Checkpoint funktionslos. Mehr zum Aufbau eines Modellordners steht in [Checkpoints, Modelle und VAE](#).

Was sind LyCORIS, LoHa, LoKr und DoRA?

Das sind Weiterentwicklungen desselben Grundgedankens, die dir beim Herunterladen begegnen. LyCORIS ist der Sammelname für eine Familie solcher Verfahren. LoHa und LoKr zerlegen die Änderung mathematisch anders – über Hadamard- beziehungsweise Kronecker-Produkte – und können bei gleichem Speicherbedarf feinere Details aufnehmen, brauchen dafür aber etwas mehr Rechenzeit beim Training. DoRA zerlegt die Anpassung zusätzlich in Betrag und Richtung und liefert damit bei kleinem Rang bessere Ergebnisse als klassisches LoRA. In der Anwendung ändert sich für dich nichts: Du lädst sie genauso und stellst dasselbe Weight ein.

Was kostet es, ein eigenes LoRA zu trainieren?

An Rechenzeit: ein bis zwei Stunden auf einer Grafikkarte mit 24 GB Speicher, also auf einer gehobenen Consumer-Karte. Bei gemieteter Rechenzeit bewegt sich das im einstelligen Eurobereich pro Lauf. Der eigentliche Aufwand steckt im Datensatz: zwanzig bis vierzig Bilder aussuchen, aufbereiten und beschriften kostet einen halben bis ganzen Arbeitstag. Wer schon eine gepflegte Bildablage hat, ist deutlich schneller.

Darf ich mit LoRAs erzeugte Bilder kommerziell nutzen?

Das hängt von drei Dingen gleichzeitig ab: der Lizenz des Basismodells, den Erlaubnissen des LoRA und der Rechtelage des Trainingsmaterials. Die restriktivste dieser drei Bedingungen bestimmt, was geht. Konkret wichtig: Die offenen Gewichte von FLUX.2 [dev] sind nicht-kommerziell lizenziert, Stable Diffusion 3.5 ist kostenfrei kommerziell nur bis eine Million US-Dollar Jahresumsatz, während Qwen-Image, Z-Image und FLUX.2 [klein] unter Apache 2.0 stehen und kommerzielle Nutzung erlauben. Prüfe das vor dem ersten Training, nicht nach der ersten Kampagne.

Brauche ich für ein LoRA einen eigenen Rechner?

Zum Anwenden nicht – ein LoRA zu laden kostet praktisch nichts zusätzlich, das läuft überall, wo auch das Basismodell läuft. Zum Trainieren brauchst du entweder eine Karte mit 24 GB Speicher oder gemietete Rechenzeit. Welcher Weg für dich sinnvoller ist, hängt an Menge, Vertraulichkeit und Taktung; die Abwägung steht in [Cloud vs. lokale KI-Bildgenerierung](#).

Fazit

Ein LoRA ist der kleine Aufsatz, der aus einem Allzweckmodell einen Spezialisten macht. Technisch ist es nichts weiter als zwei schmale Matrizen, deren Produkt auf die eingefrorenen Gewichte des Basismodells addiert wird – ein Verfahren, das für Sprachmodelle erfunden wurde und in der Bildgenerierung sein zweites, größeres Leben gefunden hat. Praktisch ist es das Werkzeug, das den Unterschied zwischen „schöne Bilder“ und „unsere Bilder“ ausmacht.

Für die tägliche Arbeit merkst du dir drei Dinge. Erstens: Das Weight ist der Regler, der alles entscheidet, und der brauchbare Bereich liegt meistens unter dem, was das Eingabefeld hergibt. Zweitens: Ein LoRA gehört zu genau einer Modellfamilie, und wenn es nichts zu tun scheint, liegt es fast immer daran oder am fehlenden Auslösewort. Drittens: Der Hebel liegt nicht bei heruntergeladenen LoRAs, sondern bei selbst trainierten – dort, wo du Datensatz, Rechte und Look in der Hand hast.

Und wenn du nur eine Sache aus diesem Beitrag mitnimmst: Kläre die Lizenz des Basismodells, bevor du den ersten Datensatz zusammenstellst. Ein LoRA lässt sich nicht auf eine andere Familie umziehen. Der Trainingslauf ist billig, der Datensatz ist es nicht.

Weiterlesen

- › Ohne passendes Basismodell läuft kein LoRA: [Checkpoints, Modelle und VAE](#)
- › Testreihen brauchen einen festen Startpunkt: [Der Seed bei der KI-Bildgenerierung](#)
- › Beschleuniger-LoRAs greifen in Schrittzahl und Guidance ein: [Sampling-Steps erklärt](#) und [CFG-Scale: Guidance in KI-Bildgeneratoren](#)
- › Detail-LoRAs mildern bekannte Modellschwächen: [Warum KI keine Hände kann](#)
- › Was in Prompt und Negativ-Prompt gehört: [Prompt und Negativ-Prompt in der KI-Bildgenerierung](#)
- › Markenkonsistenz als Ziel, nicht als Zufall: [Markenkonsistente KI-generierte Bilder](#)
- › Produktion in Serie aufsetzen: [Aufbau einer skalierbaren Bild-Maschine mit ComfyUI und Flux](#)
- › Alle Grundlagen im Überblick: [KI-Bildgenerierung](#)