Aus Rauschen wird Bild (Diffusion)

Zerstöre ein Bild in 20 Rausch-Schritten — und sieh dann, wie ein Entrausch-Schritt, 20-mal angewendet, aus purem Grieß wieder ein Motiv formt. Plus: was der Prompt dabei wirklich tut.

Kapitel online

Aktuell · Kapitel 33 von 50 · ca. 15 Min. · 250 XP

Einfach gesagt: Vor deinen Augen wird ein Bild erst zu reinem Rauschen — dann räumt die KI es Schritt für Schritt wieder weg, bis ein Motiv entsteht.

Midjourney, Stable Diffusion, DALL-E: Alle großen Bild-KIs arbeiten nach demselben, fast frech simplen Prinzip — **Diffusion**. Es besteht aus einer leichten und einer schweren Hälfte. Die leichte kannst du gleich selbst erledigen: **ein Bild kaputt machen.**

Was du hier lernst

- Wie aus einem Bild **Rauschen** wird — vorwärts, ganz ohne KI
- Warum ein Modell nur **EINEN Entrausch-Schritt** lernt und ihn 20-mal anwendet
- Was der **Prompt** dabei tut: er lenkt jeden einzelnen Schritt
- Wie so ein Schritt **runter auf Tensor-Ebene** aussieht — vier Pixel zum Mitrechnen
- Warum moderne Bild-KIs im **Latent Space** entrauschen statt auf Pixeln

Für den Unterricht

Einstieg (3 Min): Kapitel gemeinsam lesen, Alltagsbeispiel aufgreifen.

Erkunden am Gerät (9 Min): die interaktiven Module auf herrki.de/340_diffusion ausprobieren.

Sichern mit Quiz (3 Min): „Kurzer Check" besprechen, „Das nimmst du mit" festhalten.

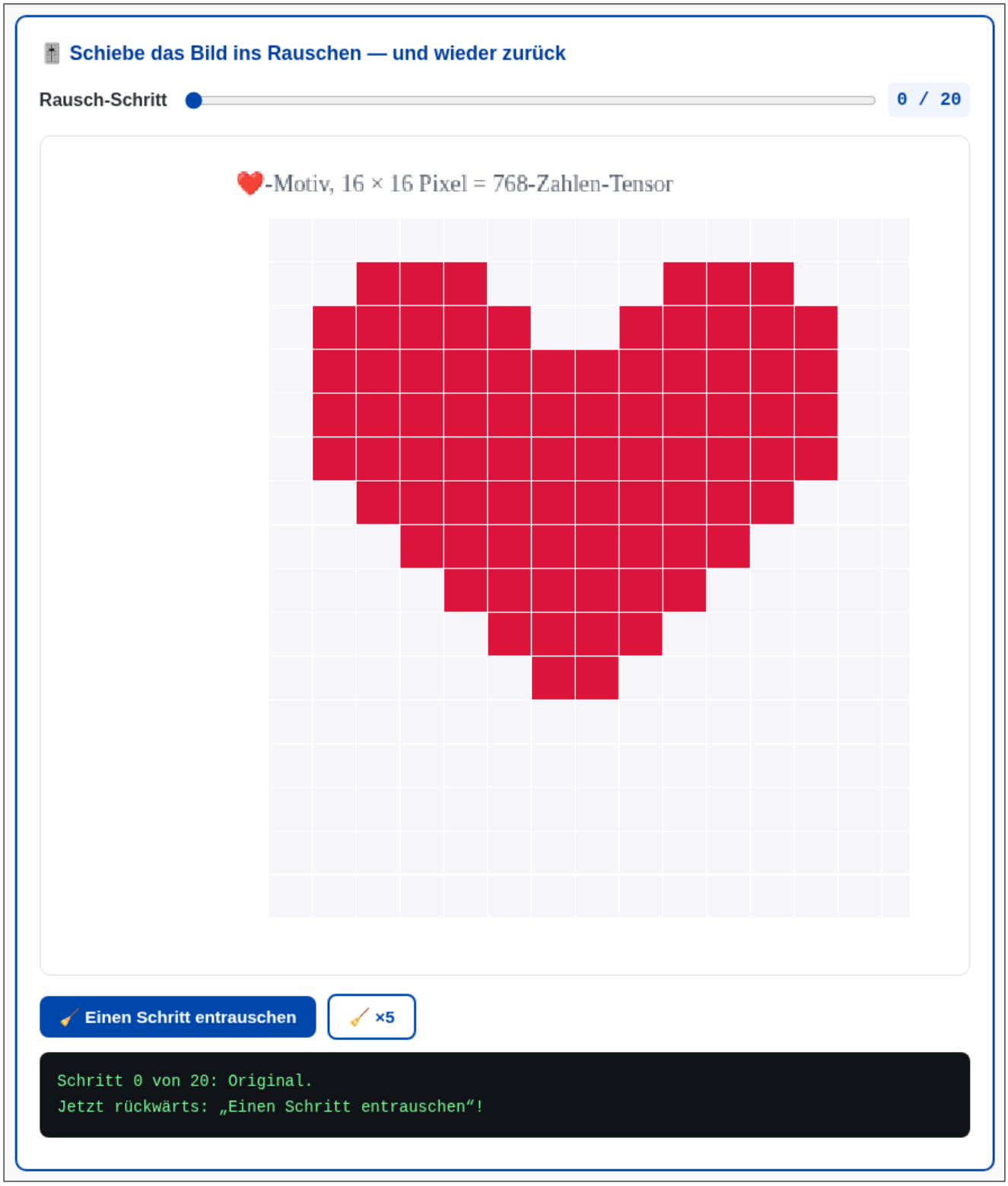
Vorwärts: Bild Rauschen (kinderleicht)

Denk dabei an „Bilder als Tensoren“: Ein Bild ist für den Rechner nichts als ein Würfel voller Zahlen. „Rauschen dazumischen“ heißt also schlicht — auf jede dieser Zahlen eine Zufallszahl draufrechnen.

Nimm ein Bild und mische in 20 Schritten immer mehr Zufallszahlen hinein, bis nur noch Grieß übrig ist. Das kann jeder Taschenrechner — dafür braucht niemand eine KI:

Am Gerät ausprobieren: herrki.de/340_diffusion#abschnitt-2

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

Merkst du den Unterschied der Richtungen? Vorwärts (Regler nach rechts) ist stumpfes Zufalls-Mischen. **Rückwärts** (Entrauschen-Knopf) muss aus dem Grieß erraten,

wo Kanten und Flächen waren — die ersten Schritte bringen viel Struktur zurück, der letzte Feinschliff ist am schwersten. Genau DIESE Richtung ist das, was eine Bild-KI lernt.

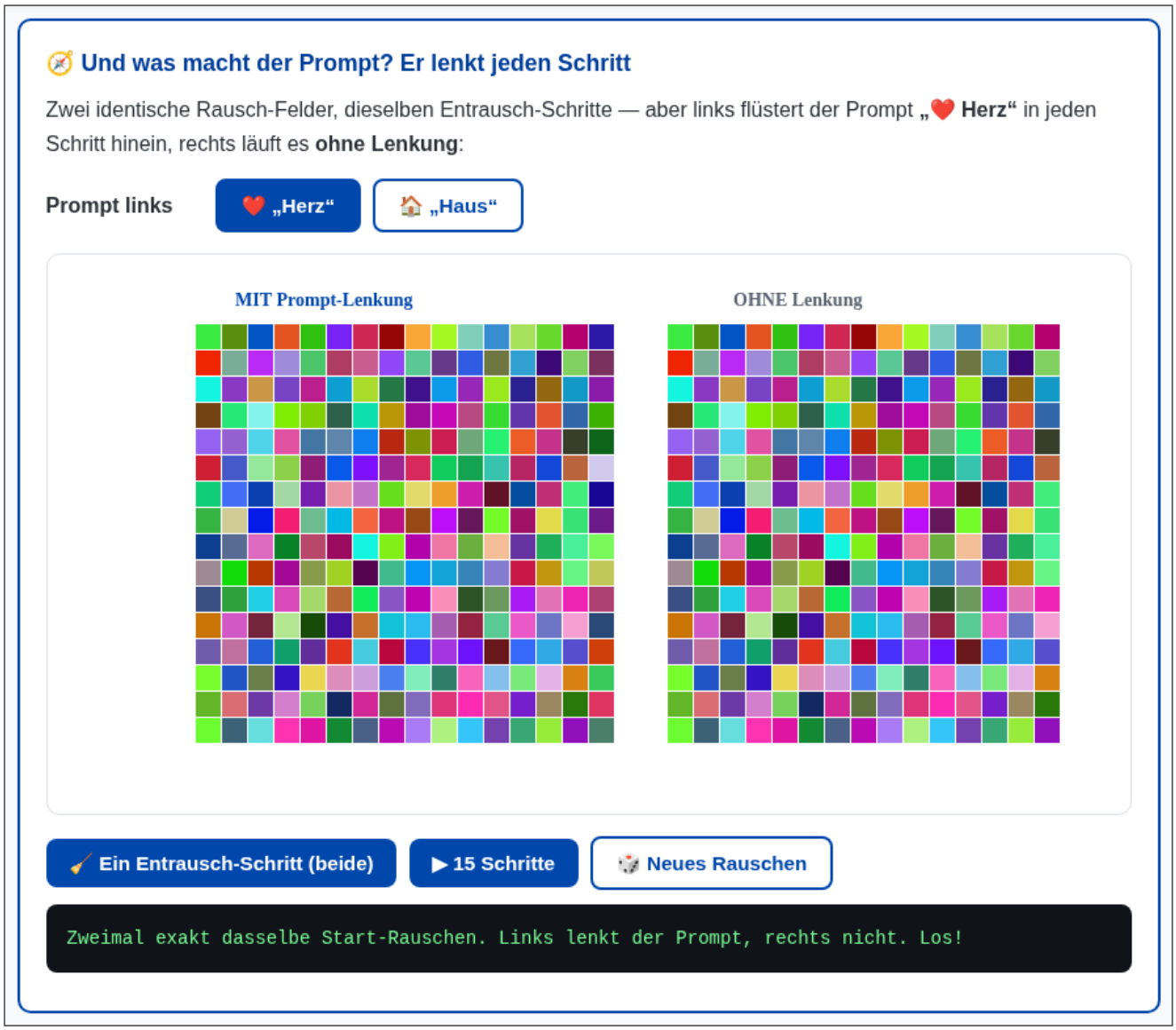
Der Trick: Ein Modell lernt EINEN Schritt — und wird 20-mal angewendet

Beim Training bekommt das Netz millionenfach Paare gezeigt: *„Hier ein verrauschtes Bild — sag mir, welches Rauschen drinsteckt.“* Es lernt also nur eine einzige Fähigkeit: **einen Schritt Rauschen abziehen**. Bei der Bild-Erzeugung startet man dann mit purem Zufalls-Rauschen und wendet diesen einen gelernten Schritt 20-, 30-, 50-mal hintereinander an. Aus Nichts entsteht ein Bild.

Neu ist daran nichts: Das ist ein Netz wie in „Was ist ein Neuronales Netz?“, das seine Vorhersage mit dem tatsächlich eingemischten Rauschen vergleicht und diesen Abstand — den Loss — Schritt für Schritt kleiner macht. Nur die Aufgabe ist ungewohnt: Vorhergesagt wird kein Preis, sondern ein Bild voller Zufallszahlen.

Am Gerät ausprobieren: herrki.de/340_diffusion#abschnitt-3

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

⚠ Ehrlichkeits-Hinweis: Diese Demo ist eine stark vereinfachte Veranschaulichung des Prinzips (unser „Modell“ ist ein kleiner Filter + Lenkung). Echte Diffusionsmodelle sagen das Rauschen mit einem trainierten neuronalen Netz voraus — der Ablauf *Rauschen → gelenktes schrittweises Entrauschen → Bild* ist aber exakt derselbe.

Runter zum Tensor

Du kennst alle Bausteine schon — also rechnen wir einen Schritt mit der Hand nach. Wir nehmen dafür nur **vier Pixel** aus dem Motiv (zum Mitrechnen verkleinert — ein 512×512-Farbbild sind in echt 786.432 Zahlen). Jede Zahl ist eine Helligkeit von 0 = schwarz bis 1 = weiß:

```
// VORWAERTS — genau das macht dein Regler bei Schritt 4 von 20, also t = 0.20
Motiv-Pixel [4]: 0.620 0.300 0.850 0.440 ← vier Graustufen aus dem Bild
Rausch-Pixel [4]: 0.910 0.140 0.070 0.680 ← ein Wuerfelwurf pro Pixel

// verrauscht = bild · (1 − t) + rauschen · t
Pixel 1: 0.620·0.80 + 0.910·0.20 = 0.496 + 0.182 = 0.678
Pixel 2: 0.300·0.80 + 0.140·0.20 = 0.240 + 0.028 = 0.268
Pixel 3: 0.850·0.80 + 0.070·0.20 = 0.680 + 0.014 = 0.694

Verrauscht [4]: 0.678 0.268 0.694 0.488 ← genau das zeigt dir das Canvas oben
```

Das war die leichte Hälfte: vier Multiplikationen, vier Additionen. Jetzt die schwere Richtung. Das Netz bekommt **nur die verrauschte Zeile** zu sehen und soll sagen, welches Rauschen darin steckt. Es rät gut — aber nie perfekt:

```
// RUECKWAERTS — ein einziger Entrausch-Schritt
Netz schaetzt: 0.890 0.160 0.110 0.660 ← geschaetzt = modell.predict(verrauscht)
Wirklich war es: 0.910 0.140 0.070 0.680 ← dicht dran, aber daneben

// abziehen und wieder auf volle Staerke bringen: (verrauscht − t·geschaetzt) / (1 − t)
Pixel 1: (0.678 − 0.20·0.890) / 0.80 = 0.500 / 0.80 = 0.625
Pixel 2: (0.268 − 0.20·0.160) / 0.80 = 0.236 / 0.80 = 0.295

Wieder da [4]: 0.625 0.295 0.840 0.445
Original war [4]: 0.620 0.300 0.850 0.440 ← fast zurueck, nur nicht ganz
```

Dieses „nur nicht ganz“ ist kein Makel, sondern der Grund für die vielen Schritte: Jeder Durchlauf holt ein Stück Struktur zurück, der nächste bügelt nebenbei den Schätzfehler des vorigen aus. Zwanzigmal hintereinander — und aus Grieß wird ein Bild.

Und so heißt genau diese Rechnung im Code. Statt vier Pixeln steht dort der ganze Bild-Tensor, Form [512, 512, 3] — die Arithmetik pro Zahl ist aber Zeile für Zeile dieselbe wie oben:

```
// Ein Bild ist ein Tensor (Kapitel 020). Rauschen ist ein Zufalls-Tensor gleicher Form.
verrauscht = bild.mul(1 - t).add( rauschen.mul(t) ) // vorwaerts: dein Regler
// rueckwaerts lernt das Netz: sage das Rauschen voraus …
geschaetzt = modell.predict(verrauscht)
// … und zieh es ein Stueck ab. 20x wiederholen -> Bild.
```

Und jetzt der Bogen zu „Latent Space“: Moderne Bild-KIs entrauschen **nicht die Pixel selbst**, sondern die komprimierte Latent-Version des Bildes — bei Stable Diffusion z. B.

64×64×4 = **16.384 Zahlen statt 786.432 Pixel-Werte** (rund 48-mal weniger). Deshalb schafft dein Grafikchip das in Sekunden.

Zurück zur Anwendung

Wenn du das nächste Mal einer Bild-KI beim „Malen“ zusiehst — dieses Schärferwerden von grobem Matsch zu feinen Details — dann siehst du in Echtzeit genau deine Entrausch-Schleife von oben: Rauschen rein, Prompt lenkt, Schritt für Schritt Richtung Motiv. Keine Magie. Ein gelernter Schritt, oft wiederholt.

Zum Weiterlesen

- **Das Paper, mit dem Diffusion praktisch wurde:** Jonathan Ho u. a., *„Denoising Diffusion Probabilistic Models“*, 2020 — arxiv.org/abs/2006.11239 — genau die Entrausch-Schleife von oben, sauber hergeleitet
- **Warum im Latent Space entrauscht wird:** Robin Rombach u. a., *„High-Resolution Image Synthesis with Latent Diffusion Models“*, 2021 — arxiv.org/abs/2112.10752 — das Paper hinter Stable Diffusion und den 16.384 statt 786.432 Zahlen
- **Die freundliche Herleitung:** Lilian Weng, *„What are Diffusion Models?“* — lilianweng.github.io (lilianweng.github.io/posts/2021-07-11-diffusion-models) — Schritt für Schritt von der Idee bis zur Formel, mit Grafiken
- **Die Ursprünge:** Jascha Sohl-Dickstein u. a., *„Deep Unsupervised Learning using Nonequilibrium Thermodynamics“*, 2015 — arxiv.org/abs/1503.03585 — fünf Jahre vor dem Hype: Bilder kaputt machen und den Weg zurück lernen

Kurzer Check – mit Lösung

Lösungen für Lehrkräfte – für Lernende die Online-Version nutzen.

Frage 1: Was lernt ein Diffusionsmodell im Training?

- Fertige Bilder auswendig, die es später heraussucht.
- **✓ Aus einem verrauschten Bild das enthaltene Rauschen vorherzusagen — also EINEN Entrausch-Schritt.**
- Die Namen aller berühmten Maler.

Nur dieser eine Schritt wird gelernt — und bei der Erzeugung viele Male hintereinander angewendet, beginnend bei purem Zufalls-Rauschen.

Frage 2: Welche Rolle spielt der Text-Prompt beim Bild-Erzeugen?

- **✓ Er lenkt jeden einzelnen Entrausch-Schritt in Richtung des beschriebenen Motivs.**
- Er wird als Wasserzeichen ins Bild geschrieben.
- Er wählt ein fertiges Bild aus einer Datenbank.

Ohne Lenkung entsteht auch „irgendetwas“ — aber eben Beliebiges/Matschiges. Der Prompt schiebt bei jedem Schritt sanft Richtung Motiv, wie in deiner Zwei-Felder-Demo.

Frage 3: Warum arbeiten moderne Bild-KIs im Latent Space statt direkt auf Pixeln?

- Weil Pixel urheberrechtlich geschützt sind.
- **✓ Weil die komprimierte Darstellung ~48-mal weniger Zahlen hat — jeder der vielen Entrausch-Schritte wird dadurch drastisch billiger.**
- Weil im Latent Space kein Rauschen existiert.

16.384 statt 786.432 Zahlen pro Schritt (Kapitel 310!) — bei 20–50 Schritten pro Bild summiert sich das gewaltig. Am Ende übersetzt ein Decoder die Latent-Version zurück in Pixel.

Das nimmst du mit

- **Die Vorwärts-Richtung ist trivial.** Rauschen auf ein Bild kippen kann jeder — dafür braucht es keine KI, und genau daraus entstehen die Trainingsbeispiele.
- **Gelernt wird genau EIN Schritt.** „Welches Rauschen steckt hier drin?“ — und dieser eine Schritt wird 20- bis 50-mal hintereinander angewendet.
- **Der Prompt lenkt jeden Schritt.** Ohne Lenkung entsteht auch etwas, nur eben Beliebiges; der Text schiebt jedes Mal sanft Richtung Motiv.
- **Gerechnet wird im Latent Space.** 16.384 statt 786.432 Zahlen pro Schritt — rund 48-mal weniger, deshalb läuft Bild-KI auf normalen Grafikkarten.
- **Nichts wird herausgesucht.** Am Anfang steht Zufalls-Rauschen, am Ende ein Bild, das es vorher nicht gab — keine Datenbank, kein Zusammenkleben.

Weiter geht’s: Eine Bild-KI rät das Rauschen weg, ein Sprachmodell rät das nächste Wort. Warum dabei manchmal etwas herauskommt, das großartig klingt und trotzdem frei erfunden ist, siehst du in „Warum KI überzeugend Unsinn erzählt“.

So zitierst du dieses Kapitel

HerrKI: Aus Rauschen wird Bild (Diffusion). herrki.de/340_diffusion, ein privates Hobbyprojekt von Terragon.de, abgerufen am 09.09.2026.