Vom Autoencoder zur physischen KI

am Beispiel des automatisierten Fahrzeugs

Eine anschauliche Einführung für Nicht-Experten

Juli 2026

MCG Management Consult GmbH

Inhaltsverzeichnis

Einleitung

Generative Künstliche Intelligenz ist im Alltag angekommen: Programme schreiben Texte, malen Bilder auf Zuruf und erzeugen Videos, die von Gefilmtem kaum noch zu unterscheiden sind. Von außen wirkt das wie Zauberei – man tippt einen Satz, und eine Maschine erschafft etwas, das es vorher nicht gab. Dieses Dokument öffnet die Blackbox. Die eigentliche Überraschung dabei: Hinter der Zauberei steckt kein unbegreifliches Formelgebirge, sondern eine Handvoll verblüffend einfacher Ideen, die aufeinander aufbauen – und jede davon lässt sich mit Alltagsbildern erklären: mit Stichpunkten und Zusammenfassungen, mit Landkarten, einem Kunstfälscher und seinem Prüfer, einem Polaroid, das sich aus dem Grieseln entwickelt. Vorwissen ist nicht nötig; jedes Fachwort wird genau dort eingeführt, wo es zum ersten Mal gebraucht wird.

Nur, warum beginnt eine Geschichte über das Erschaffen ausgerechnet mit einem so sperrigen Wort wie „Autoencoder“ – einer Maschine, die nichts weiter tut, als ihre Eingabe zu kopieren? Weil sie der gemeinsame Vorfahr der ganzen Familie ist. Im scheinbar sinnlosen Kopieren steckt bereits der Kern von allem, was danach kommt: verdichten, das Wesentliche vom Verzichtbaren trennen, wiederherstellen. Jedes weitere Kapitel ist eine konsequente Weiterentwicklung dieser einen Idee – vom Kopieren zum Erschaffen, vom Bild zum Video, vom Video zur begehbaren Welt, von der begehbaren Welt zum automatisierten Fahrzeug. Wer Kapitel 1 verstanden hat, hält deshalb den Schlüssel für alles Weitere in der Hand; der Rest der Reihe baut nur noch an.

Zur Orientierung: Dieses Dokument fasst eine siebenteilige, für Nicht-Experten verständliche Reihe von Erläuterungen zur generativen Künstlichen Intelligenz zusammen – vom Autoencoder (Daten verdichten und wiederherstellen) über den Variational Autoencoder (VAE; sinngemäß: ein Autoencoder, der Regionen statt fester Punkte lernt), Generative Adversarial Networks (GANs; sinngemäß: ein generatives Netz, im Wettstreit mit einem Prüfnetz trainiert) und Diffusionsmodelle (Erzeugen durch schrittweises Entrauschen) bis zu den aktuellen Entwicklungen und Zukunfts-erwartungen rund um Denkzeit-Skalierung und Weltmodelle. Kapitel 6 überträgt die Bausteine anschließend auf die physische KI im automatisierten Fahrzeug, und Kapitel 7 ergänzt die Rollen der KI rund um die Fahrfunktion – vom Sprachmodell an Bord über die lernende Flotte bis zum Entwicklungsprozess.

Die Kapitel bauen aufeinander auf und verwenden durchgängig dieselbe Bildsprache: die Codekarte (violett), die arbeitenden Netze (grün), Zufallspunkte (gelb-orange), Lernsignale (bernsteinfarbene Kästen), der Prüfer (rosa), geprüfte, sichere Zustände (hellgrün) sowie Warnungen und Rückfallebene (lachsrot). Wer von vorn liest, begegnet denselben Bausteinen immer wieder – bis hin zur doppelten Pointe: In praktisch jedem modernen Bildgenerator arbeitet wörtlich noch ein Autoencoder weiter, und am Ende verlassen die Bausteine den Bildschirm und bringen ein reales Fahrzeug sicher durch den Verkehr.

1. Der Autoencoder: Komprimieren als Lernprinzip

Die Grundidee

Autoencoder gehören zu den elegantesten Ideen im maschinellen Lernen, weil sich das Prinzip mit einer Alltagssituation erklären lässt. Stell dir vor, jemand liest einen langen Artikel und darf sich nur drei Stichpunkte notieren. Eine zweite Person muss den Artikel dann allein aus diesen Stichpunkten wieder ausformulieren. Vergleicht man das Ergebnis mit dem Original, sieht man sofort, wie gut die Stichpunkte waren: Je ähnlicher die Rekonstruktion, desto besser wurde das Wesentliche erfasst. Genau das macht ein Autoencoder – nur mit Daten statt Artikeln, und beide „Personen“ sind Teile desselben neuronalen Netzes.

Ein Autoencoder besteht aus drei Teilen: Der **Encoder** verdichtet die Eingabe (z. B. ein Bild mit einer Million Pixelwerten) Schritt für Schritt auf einen winzigen Engpass, den sogenannten **Code** – vielleicht nur 32 Zahlen. Der **Decoder** versucht anschließend, aus diesen wenigen Zahlen das ursprüngliche Bild wiederherzustellen. Die Aufgabe klingt zunächst sinnlos („gib einfach aus, was reinkam“), aber der Flaschenhals in der Mitte erzwingt den eigentlichen Lerneffekt: Das Netz *muss* entscheiden, welche Information wesentlich ist – alles andere passt nicht hindurch.

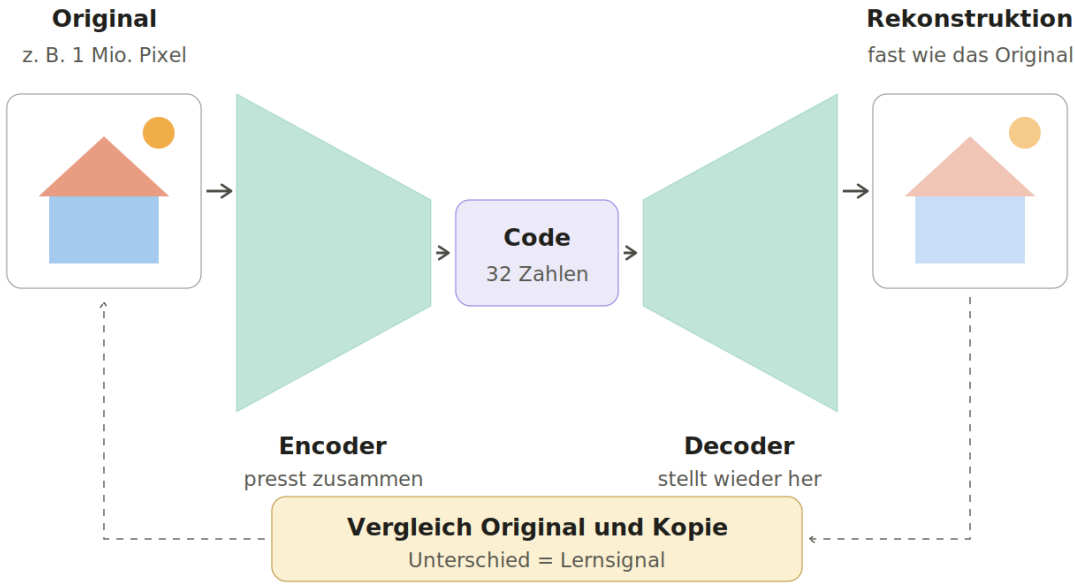

Abbildung 1: Aufbau eines Autoencoders: Encoder, Code (Flaschenhals) und Decoder – der Vergleich von Original und Kopie liefert das Lernsignal.

Das Lernen funktioniert dabei ohne jede menschliche Beschriftung der Daten – das ist einer der großen Vorteile. Das Netz vergleicht nach jedem Versuch einfach seine Rekonstruktion mit dem Original, misst den Unterschied (den *Rekonstruktionsfehler*) und justiert Encoder und Decoder gemeinsam so nach, dass dieser Fehler kleiner wird. Nach tausenden Wiederholungen hat es gelernt, seine Trainingsdaten erstaunlich gut durch den Engpass zu bekommen. Der Code in der Mitte ist dann eine Art gelernte Essenz der Daten – bei Gesichtern stecken darin sinngemäß Dinge wie Kopfhaltung, Beleuchtung oder Gesichtsform, ohne dass jemand dem Netz diese Begriffe je beigebracht hätte.

Wozu Autoencoder nützlich sind

Drei typische Anwendungen: **Datenkompression** (die verdichtete Darstellung braucht viel weniger Speicher), **Entrauschen** (trainiert man das Netz darauf, aus verrauschten Bildern saubere zu rekonstruieren, lernt es, was „sauber“ bedeutet) – und die vielleicht wichtigste: **Anomalieerkennung**. Die Logik dahinter ist verblüffend einfach: Ein Autoencoder, der nur mit normalen Beispielen trainiert wurde, kann auch nur Normales gut rekonstruieren. Zeigt man ihm etwas, das er so nie gesehen hat – eine ungewöhnliche Kreditkartentransaktion, ein untypisches Maschinengeräusch, ein seltsames Kamerabild –, misslingt die Kopie, und der Rekonstruktionsfehler schießt in die Höhe. Genau dieser hohe Fehler ist das Alarmsignal.

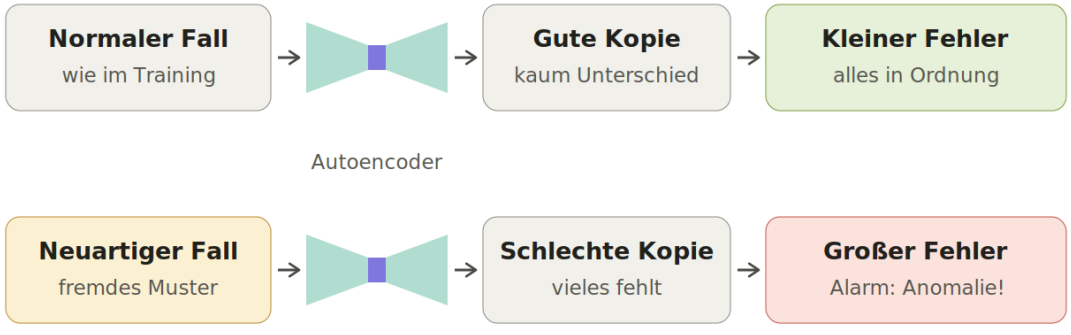

Abbildung 2: Anomalieerkennung mit einem Autoencoder: Normales wird gut rekonstruiert (kleiner Fehler), Neuartiges schlecht (großer Fehler = Alarm).

Zwei Dinge sind noch gut zu wissen. Erstens: Die Kompression ist verlustbehaftet – die Rekonstruktion ist nie perfekt, sondern eine „Erinnerung“ des Netzes an das Wesentliche. Zweitens gibt es eine berühmte Weiterentwicklung, den *Variational Autoencoder* (VAE): Er ordnet den Code so, dass man dort auch neue Punkte „erfinden“ und daraus völlig neue, plausible Bilder erzeugen kann – damit wurde der Autoencoder zu einem der Urahnen der heutigen generativen KI. Ihm ist das nächste Kapitel gewidmet.

Merksatz: Ein Autoencoder ist ein neuronales Netz, das lernt, Daten durch einen Flaschenhals zu pressen und wiederherzustellen – und dabei nebenbei lernt, was an den Daten wesentlich, was verzichtbar und was ungewöhnlich ist.

2. Der Variational Autoencoder: von der Kopie zur Neuschöpfung

Der Trick: Region statt Punkt

Der Variational Autoencoder (VAE) ist die logische Fortsetzung der Geschichte, und der Ausgangspunkt ist eine Schwäche des gewöhnlichen Autoencoders. Stell dir den Code-Raum als eine Landkarte vor: Jedes Trainingsbild bekommt vom Encoder einen exakten Punkt auf dieser Karte zugewiesen. Das Problem: Zwischen diesen Punkten liegt Niemandsland. Der Decoder hat nur gelernt, die tatsächlich besuchten Punkte zurückzuverwandeln – wählt man einen beliebigen Punkt dazwischen, kommt meist Pixelmatsch heraus. Zum *Erzeugen* neuer Bilder taugt die Karte also nicht.

Der Variational Autoencoder repariert genau das, mit einem überraschend kleinen Trick: Der Encoder liefert **nicht mehr einen Punkt, sondern eine kleine Region** – genauer gesagt eine Mitte und eine Streuung (für Fortgeschrittene: einen Mittelwert und eine Standardabweichung). Aus dieser Region wird bei jedem Trainingsdurchlauf ein **Zufallspunkt** gezogen, und erst der wandert in den Decoder.

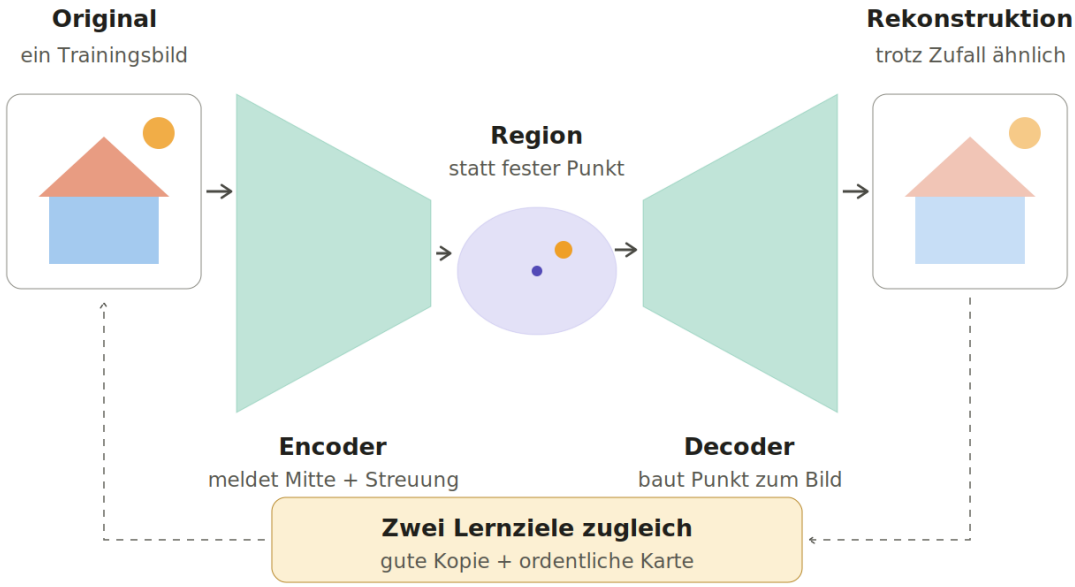

Abbildung 3: Der VAE: Der Encoder liefert eine Region (Mitte + Streuung); ein daraus gezogener Zufallspunkt wandert in den Decoder.

Warum hilft diese eingebaute Unschärfe? Weil der Decoder nie genau weiß, welcher Punkt aus der Region gezogen wird, muss er für *alle* Punkte in der Nachbarschaft ein ähnlich gutes Bild liefern. Benachbarte Codes können also nicht mehr für völlig verschiedene Dinge stehen – die Karte wird zwangsläufig glatt und zusammenhängend. Dazu kommt das zweite Lernziel aus dem Schaubild, eine Art Ordnungsregel (technisch: die KL-Divergenz): Sie zieht alle Regionen sanft zu einem gemeinsamen, standardisierten Bereich um die Kartenmitte zusammen, sodass sie sich überlappen und keine Löcher entstehen. Beide Ziele stehen übrigens in einem Spannungsverhältnis – perfekte Kopien *oder* eine perfekt aufgeräumte Karte, das Training sucht den besten Kompromiss.

Der Lohn: eine Karte ohne Löcher

Der Lohn dieser Mühe zeigt sich, wenn man die beiden Codekarten nebeneinanderlegt: Beim gewöhnlichen Autoencoder führt ein willkürlich gewählter Punkt meist ins Leere, beim VAE liefert *jeder* Punkt aus dem geordneten Bereich ein neues, plausibles Bild – aus dem Rekonstruktions-Werkzeug ist ein Erzeugungs-Werkzeug geworden.

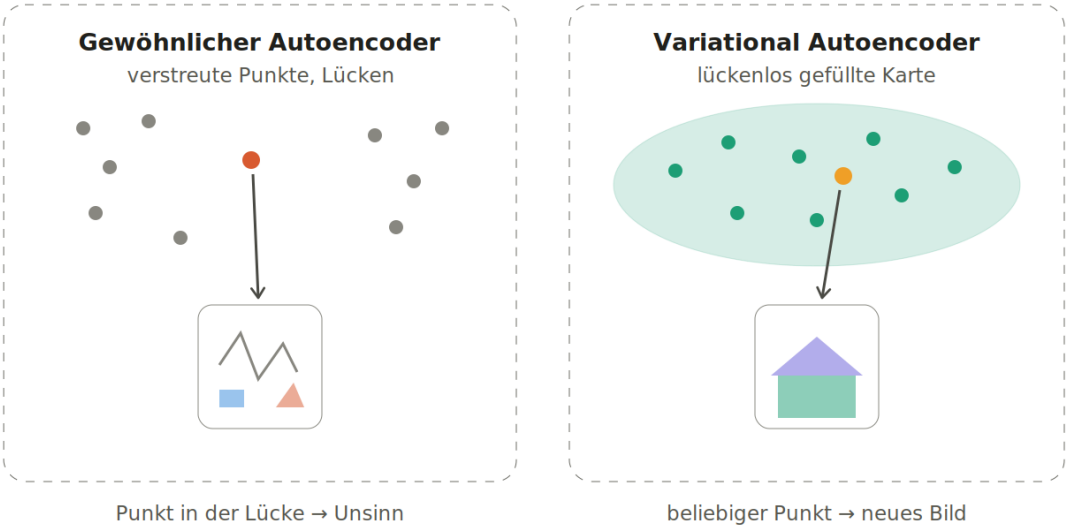

Abbildung 4: Codekarte im Vergleich: verstreute Punkte mit Lücken (gewöhnlicher Autoencoder) gegenüber der lückenlos gefüllten Karte des VAE.

Die glatte Karte hat noch einen schönen Nebeneffekt: Man kann zwischen zwei Bildern *wandern*. Wählt man den Codepunkt von Gesicht A und läuft auf der Karte schrittweise zum Punkt von Gesicht B, erzeugt der Decoder unterwegs einen fließenden Übergang – jedes Zwischenbild ist ein plausibles Gesicht. Oft entsprechen sogar bestimmte Richtungen auf der Karte verständlichen Eigenschaften, etwa „mehr Lächeln“ oder „Kopf weiter gedreht“. Ehrlicherweise haben klassische VAEs auch eine bekannte Schwäche: Ihre Bilder wirken oft etwas weichgezeichnet, eben weil der Decoder für ganze Regionen „im Mittel“ gute Ergebnisse liefern muss. Neuere Verfahren wie Diffusionsmodelle haben das überwunden – bauen aber gedanklich auf genau dieser Idee eines geordneten, durchgängigen Code-Raums auf.

Merksatz: Ein VAE ist ein Autoencoder, der jedes Bild nicht auf einen Punkt, sondern auf eine kleine Zufallsregion abbildet und seine Codekarte dabei so aufräumt, dass jeder Punkt darauf ein gültiges, neues Bild ergibt – aus Komprimieren wird Erschaffen.

3. Der Weg zur generativen KI: GAN, Diffusion, Latent Diffusion

Der Weg vom VAE zu den heutigen Bildgeneratoren ist eine Geschichte in drei Etappen – und der VAE taucht am Ende auf überraschende Weise wieder auf. Zur Erinnerung, wo wir stehen: Der VAE hat die entscheidende Idee geliefert – eine geordnete Codekarte, auf der jeder Punkt ein gültiges Bild ergibt. Sein Schönheitsfehler: Die Bilder wirken weichgezeichnet.

Etappe 1: Das GAN – Fälscher gegen Prüfer

Die erste Etappe auf dem Weg zu wirklich scharfen Bildern war ein völlig anderer Ansatz, das **GAN** (Generative Adversarial Network) – am besten erklärt als Duell zwischen einem Kunstfälscher und einem Kunstprüfer. Der *Generator* (ein Verwandter unseres Decoders) bekommt wie beim VAE nur einen Zufallspunkt und soll daraus ein Bild fälschen. Ein zweites Netz, der *Prüfer*, bekommt abwechselnd echte Trainingsbilder und Fälschungen vorgelegt und muss entscheiden: echt oder falsch? Sein Urteil ist das Lernsignal für beide – der Prüfer wird misstrauischer, der Fälscher raffinierter.

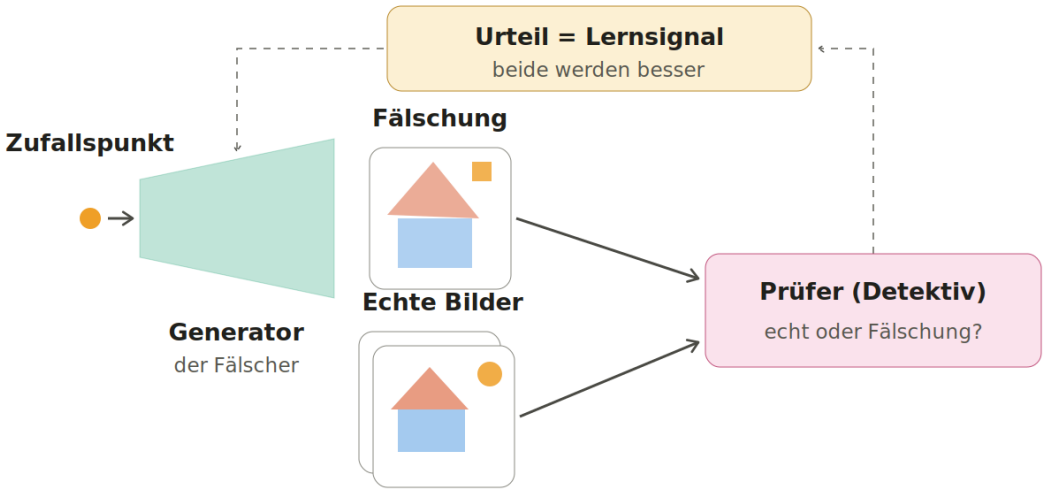

Abbildung 5: Das GAN-Prinzip: Der Generator fälscht aus einem Zufallspunkt, der Prüfer vergleicht mit echten Bildern – sein Urteil ist das Lernsignal für beide.

Dieses Wettrüsten endet im Idealfall damit, dass der Prüfer nur noch raten kann – die Fälschungen sind nicht mehr vom Original zu unterscheiden. GANs lieferten ab etwa 2018 die ersten gestochen scharfen KI-Gesichter von Menschen, die nie existiert haben. Aber das Duell ist launisch: Das Training gerät leicht aus dem Gleichgewicht, der Fälscher verlegt sich gern auf wenige „Lieblingsmotive“, und gezielt steuern lässt sich das Ganze schlecht.

Etappe 2: Diffusion – erzeugen in kleinen Schritten

Die **Diffusionsmodelle** der zweiten Etappe verfolgen deshalb eine ganz andere Philosophie: nicht ein Bild in einem großen Wurf fälschen, sondern es in vielen kleinen, verlässlichen Handgriffen aus dem Rauschen herausarbeiten. Das Training ist verblüffend simpel: Man nimmt ein echtes Bild und überstreut es Schritt für Schritt mit Rauschen, bis nur noch Bildschirmschnee übrig ist. Das Netz lernt dabei genau eine bescheidene Fähigkeit – *einen* kleinen Rauschschritt rückgängig zu machen.

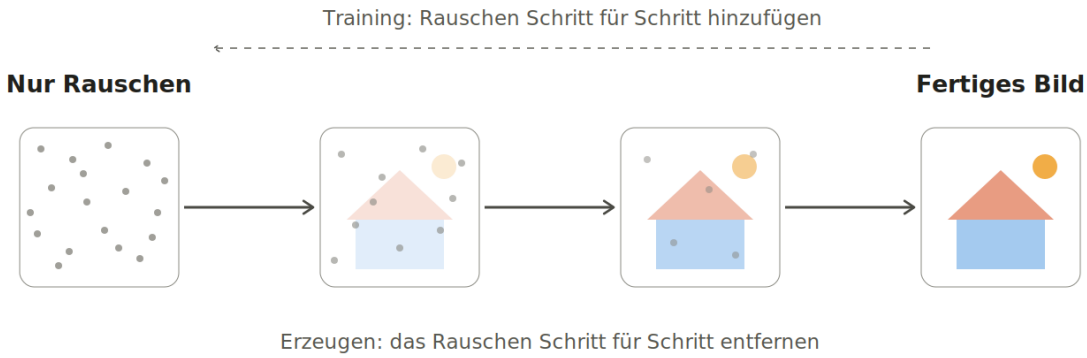

Abbildung 6: Das Diffusionsprinzip: Im Training wird Rauschen Schritt für Schritt hinzugefügt, beim Erzeugen Schritt für Schritt entfernt.

Zum Erzeugen dreht man den Spieß einfach um: Man startet mit reinem Zufallsrauschen und wendet die gelernte Entrausch-Fähigkeit einige Dutzend Mal hintereinander an – wie ein Polaroid, das sich langsam aus dem Grieseln entwickelt. Weil das Lernziel so schlicht ist, ist das Training robust und die Bildqualität spektakulär. Der Haken: Es ist langsam und teuer, denn jeder der vielen Schritte rechnet auf Millionen von Pixeln.

Etappe 3: Latent Diffusion – der VAE kehrt zurück

Und genau hier, in der dritten Etappe, kehrt der VAE zurück. Die Idee der **Latent Diffusion** (so funktioniert z. B. Stable Diffusion): Entrausche nicht das riesige Pixelbild, sondern einen winzigen Punkt auf der komprimierten VAE-Codekarte – dort ist jeder Schritt tausendfach billiger. Erst ganz am Ende übersetzt der VAE-Decoder den fertigen Code in das große, scharfe Bild. Und die Steuerung liefert der Prompt: Ein Text-Encoder übersetzt die Worte in ein Signal, das jeden einzelnen Entrauschungsschritt in die passende Region der Karte schubst.

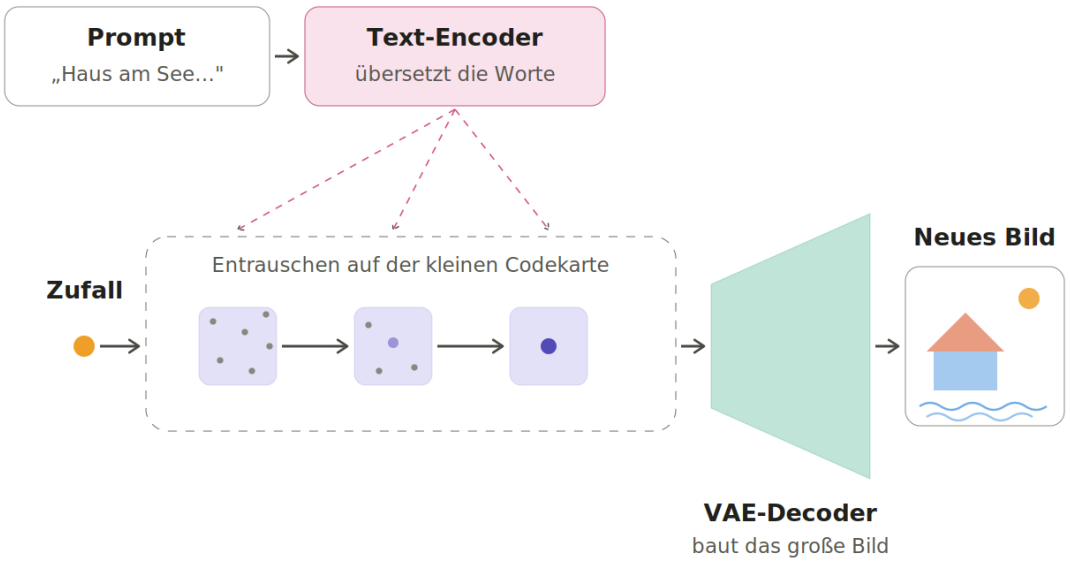

Abbildung 7: Latent Diffusion: Entrauschen auf der kleinen VAE-Codekarte, gelenkt vom Text-Encoder – der VAE-Decoder baut daraus das große Bild.

Damit schließt sich der Kreis auf elegante Weise: In praktisch jedem modernen Bildgenerator steckt wörtlich ein VAE als Ein- und Ausgangstor – die Diffusion hat nicht seinen Platz eingenommen, sondern

arbeitet *auf seiner Karte*. Der Vollständigkeit halber: Text-KI wie ChatGPT geht einen anderen Weg – dort sagt ein Transformer (das „T“ in ChatGPT) Wort für Wort das jeweils plausibelste nächste Stück voraus. Die Grundidee ist aber dieselbe: Ein Netz verinnerlicht die Struktur seiner Trainingsdaten so gründlich, dass es daraus Neues erschaffen kann, das es nie gesehen hat.

Merksatz: Der VAE lieferte die geordnete Karte, das GAN bewies im Fälscher-Duell, dass gestochen scharfe Neuschöpfungen möglich sind, die Diffusion machte das Erzeugen zum verlässlichen Handwerk in kleinen Schritten – und heutige Generatoren wie Stable Diffusion vereinen alles: Sie entrauschen, von den Worten des Prompts gelenkt, einen Punkt auf der VAE-Codekarte.

4. Vertiefung: Diffusion kompakt und der Unterschied zwischen Decoder und Generator

Diffusion kompakt: Üben mit Musterlösung

Die Diffusion und der Vergleich von Decoder und Generator gehören eng zusammen – deshalb zunächst die Diffusion kompakt, woraus sich der Vergleich fast von selbst ergibt. Der Kern der Diffusion ist eine einzige, bescheidene Fähigkeit: *ein kleines bisschen Rauschen entfernen*. Und das Training dafür ist deshalb so unschlagbar robust, weil man sich die Übungsaufgaben selbst herstellen kann: Man nimmt ein echtes Bild und streut eigenhändig Rauschen darüber. Damit liegt zu jeder Aufgabe automatisch die Musterlösung bereit – das Netz macht seinen Entrauschungsversuch, und der Vergleich mit dem Original zeigt sofort, wie gut er war. Kein Duell wie beim GAN, kein Kompromiss zwischen zwei Zielen wie beim VAE, sondern Millionen einfacher Übungsaufgaben mit beiliegender Lösung.

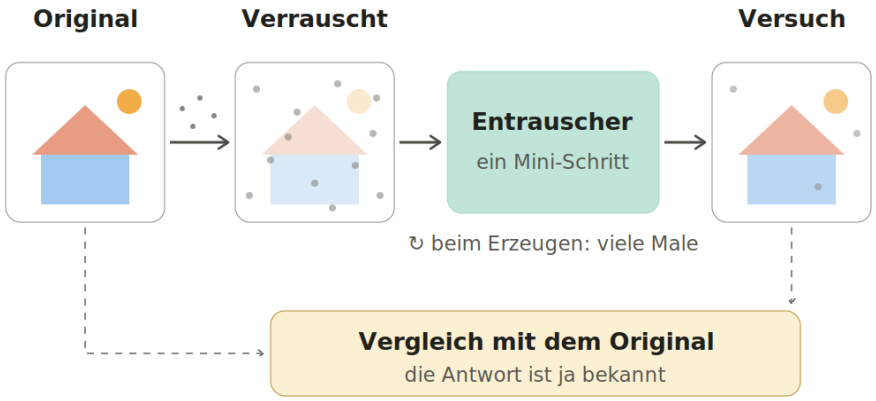

Abbildung 8: Training des Entrauschers: Zu jeder verrauschten Übungsaufgabe liegt die Musterlösung (das Original) automatisch bereit.

Beim Erzeugen wird diese eine geübte Fähigkeit dann einfach in Serie geschaltet: Man startet mit reinem Zufallsrauschen und wendet den Entrauscher einige Dutzend Mal hintereinander an – jeder Schritt muss das Bild nur ein kleines bisschen besser machen, und aus vielen verlässlichen Trippelschritten wird ein großes Kunststück.

Decoder und Generator: gleicher Bauplan, anderer Lehrer

Auf den ersten Blick sind **Decoder** und **Generator** Zwillinge. Beide sind die „malende Hälfte“ ihres Systems: Sie bekommen einen Punkt aus einem Code-Raum und verwandeln ihn in ein Bild. Der Unterschied liegt nicht im Bauplan, sondern im *Lehrer*. Der Decoder malt nach Vorlage: Sein Encoder-Partner hat ihm den Punkt aus einem konkreten Original gemacht, also kann man seine Kopie direkt daneben halten und jede Abweichung anstreichen. Das macht ihn treu, aber auch sicherheitsliebend – wo mehrere scharfe Bilder gleich gut passen würden, malt er lieber deren weichen Mittelwert. Der Generator dagegen hat gar kein Original: Für seinen rohen Zufallspunkt existiert keine „richtige“ Antwort, die man danebenlegen könnte. Sein einziger Lehrer ist das Urteil des Prüfers – *wirkt das echt?* Das zwingt ihn, sich mutig auf Details festzulegen (Weichzeichnung würde sofort auffliegen), macht sein Training aber launisch.

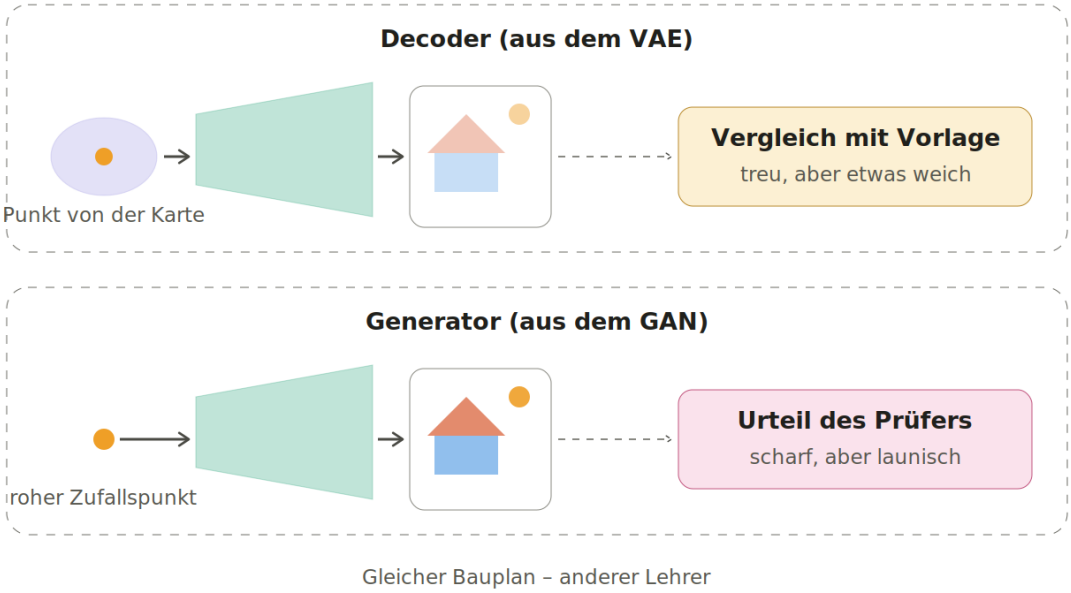

Abbildung 9: Decoder und Generator im Vergleich: gleicher Bauplan, anderer Lehrer – Vorlage-Vergleich gegenüber Prüfer-Urteil.

Und jetzt sieht man auch, warum die Diffusion so gut abschneidet: Ihr Entrauscher vereint das Beste aus beiden Welten. Er lernt wie ein Decoder – immer mit beiliegender Musterlösung, also stabil und ohne Launen. Aber er erschafft wie ein Generator – beim Erzeugen startet er aus reinem Zufall, ganz ohne Vorlage. Der Trick, der diesen Spagat möglich macht, ist die Zerlegung in viele kleine Schritte: Für einen Mini-Schritt gibt es immer eine eindeutige richtige Antwort, für ein ganzes Bild aus dem Nichts nicht.

Merksatz: Decoder und Generator sind baugleiche Maler mit verschiedenen Lehrern – der Decoder lernt *Treue* durch den Vergleich mit der Vorlage, der Generator lernt *Überzeugungskraft* durch das Urteil eines Kritikers, und der Entrauscher der Diffusion umgeht dieses Entweder-oder, indem er das Erschaffen in viele kleine Schritte zerlegt, von denen jeder eine bekannte Musterlösung hat.

5. Wie es weitergeht: schneller, kontrollierbarer, klüger (Stand: Juli 2026)

Die Verbesserung der KI-Ausgabe läuft derzeit auf drei Baustellen gleichzeitig – *schneller*, *kontrollierbarer* und *klüger* –, und am Horizont wartet ein Sprung, der über bessere Bilder hinausgeht: begehbare Welten.

Baustelle 1: Schneller – die Abkürzung über die Karte

Die klassische Diffusion braucht Dutzende Trippelschritte, und genau das ist ihr Flaschenhals: Bisher ist Bilderzeugung meist ein Stapelprozess – Anfrage abschicken, Sekunden bis Minuten warten, Ergebnis prüfen – und die nächste Grenze ist die Erzeugung in Echtzeit, bei der man eine Szene noch während des Entstehens anpassen kann. Zwei Ideen machen das möglich. Erstens die *Destillation* – wieder ein Lehrer-Schüler-Prinzip: Ein fertig trainiertes Modell läuft seine 50 Trippelschritte, und ein Schüler-Netz lernt, direkt dorthin zu *springen*, wo der Lehrer am Ende ankommt. Zweitens das *Flow Matching*: Man bringt dem Netz von vornherein möglichst gerade Wege vom Rauschen zum Bild bei, statt des historisch gewachsenen Zickzacks. Beides zusammen drückt die 50 Schritte auf 1 bis 4 – und plötzlich entstehen Bilder, während man noch tippt.

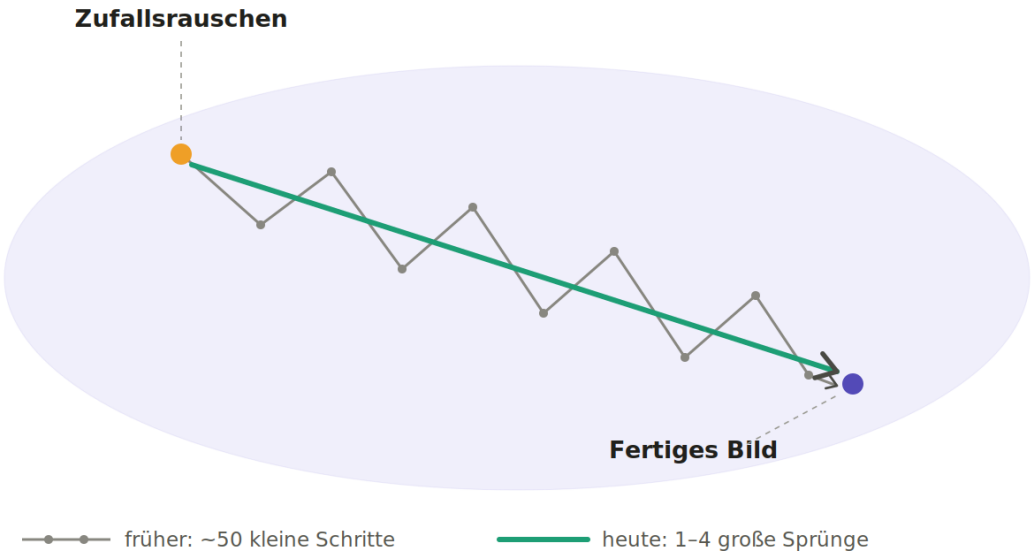

Abbildung 10: Die Abkürzung über die Codekarte: viele kleine Schritte (klassische Diffusion) gegenüber wenigen großen Sprüngen (destillierte Modelle).

Baustelle 2: Kontrollierbarer – vom Würfeln zum Werkzeug

Die zweite große Verbesserung betrifft weniger die Bildqualität als die *Vorhersagbarkeit*: Nutzer wollen zunehmend mehr Kontrolle über ihr Ausgangsmaterial – auf Videoplattformen macht die Variante „Bild zu Video“, bei der ein eigenes Startbild animiert wird, bereits rund ein Drittel aller Aufträge aus, weil sie vorhersagbarere Ergebnisse liefert als reine Textprompts. Dazu kommen Bearbeitung per Anweisung („mach den Himmel bewölkt, lass das Haus stehen“) statt Neuwürfeln, und vor allem *Konsistenz*: Die zeitliche Konsistenz – dass Figuren und Umgebungen über alle Videobilder hinweg stabil bleiben – galt lange als heiliger Gral der Videoerzeugung und ist inzwischen so weit gereift, dass KI-Inhalte oft nicht mehr von Gefilmtem zu unterscheiden sind. Wiedererkennbare Charaktere über mehrere Szenen hinweg und die Bildberechnung in Echtzeit (Rendering) zählen zu den prägenden Trends, und strukturell wachsen die einzelnen Spezialwerkzeuge gerade zu integrierten Suiten zusammen, die Bild, Video und Bearbeitung in einem Produkt vereinen.

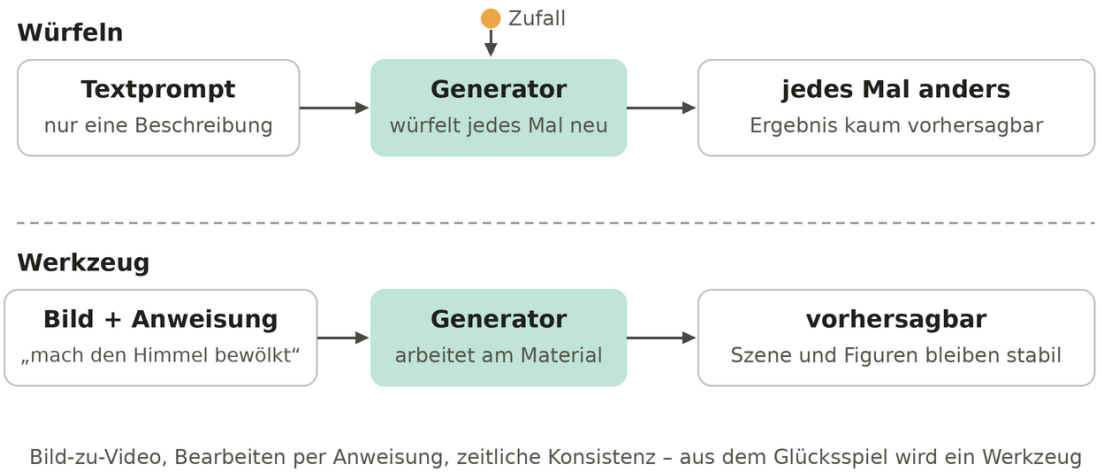

Abbildung 11: Vom Würfeln zum Werkzeug: Startbild, Anweisungen und erzwungene Konsistenz machen die Erzeugung vorhersagbar.

Baustelle 3: Klüger – Denkzeit statt nur Modellgröße

Das ist der vielleicht wichtigste neue Hebel, und er gilt für alle KI-Ausgaben, besonders für Text. Jahrelang lautete das Rezept „größeres Modell, mehr Trainingsdaten“. Weil dieses Training an Rechen- und Datengrenzen stößt, hat sich ein neues Paradigma etabliert: mehr Rechenaufwand im Moment der Antwort – das Modell denkt länger nach, probiert mehrere Lösungskandidaten, prüft und verbessert sich selbst, bevor es die endgültige Antwort ausgibt. Das gibt es in zwei Spielarten: sequenziell, indem das Modell seinen Entwurf selbst reflektiert und nachbessert, oder parallel, indem es mehrere Kandidaten gleichzeitig erzeugt und ein Prüfmodell den besten auswählt. Das Muster ist vertraut: Der *Prüfer* aus dem GAN kehrt zurück – nur ist er diesmal kein Gegner im Duell, sondern die innere Qualitätskontrolle des Modells selbst.

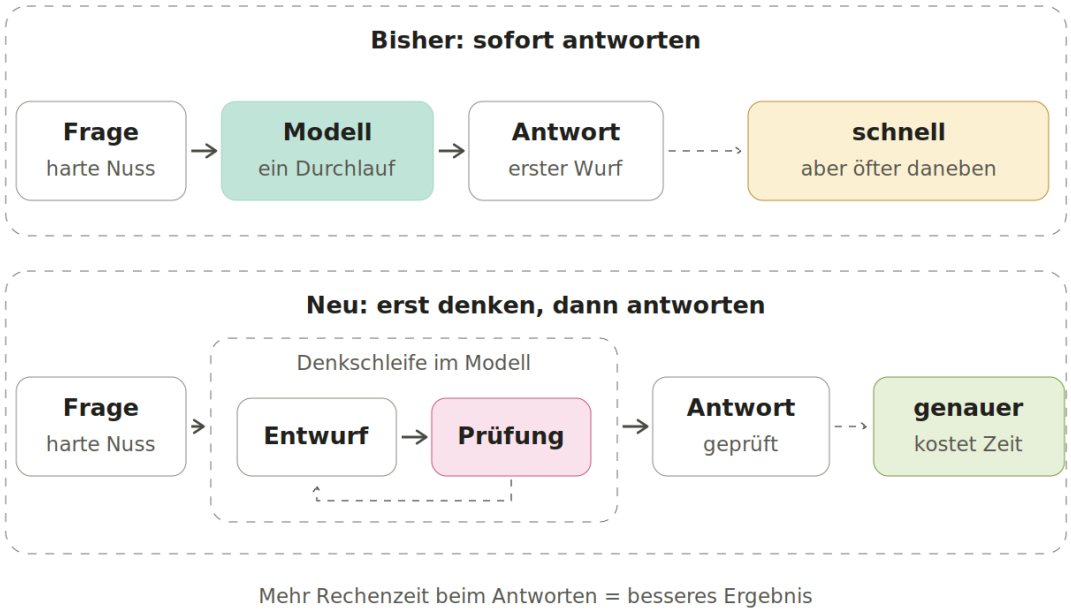

Abbildung 12: Sofort antworten oder erst denken: Die Denkschleife aus Entwurf und Prüfung liefert genauere Antworten – gegen mehr Rechenzeit.

Trainiert wird dieses Denken übrigens mit einem alten Bekannten: der Musterlösung. Man lässt das Modell an Aufgaben üben, deren Ergebnis sich objektiv nachprüfen lässt – Mathematik, Programmcode – und belohnt die Denkwege, die zur richtigen Antwort führen. Wie sehr sich das Gewicht verschiebt, zeigt die Infrastruktur: Der Rechenbedarf für das *Antworten* wird Prognosen zufolge den für das Training um mehr als das Hundertfache übersteigen, weil denkende Modelle um Größenordnungen mehr Rechenschritte pro Antwort verbrauchen.

Ausblick: Vom fertigen Video zur begehbaren Welt

Der spannendste erwartete Sprung ist der von der *Ausgabe* zur *Simulation*. Sogenannte Weltmodelle erzeugen kein fertiges Video mehr, sondern immer nur das jeweils nächste Bild – autoregressiv, als Antwort auf die laufenden Eingaben des Nutzers. Das bekannteste Beispiel ist Genie 3 von Google DeepMind: Aus einem Textprompt entsteht eine erkundbare Welt, die in Echtzeit auf Nutzereingaben reagiert – seit Ende Januar 2026 ist das als „Project Genie“ öffentlich zugänglich (zunächst für Abonnenten in den USA, seit Mai 2026 weltweit), mit 720p bei 24 Bildern pro Sekunde. Die minutenlange Konsistenz der Welt ist dabei eine Fähigkeit des zugrunde liegenden Genie-3-Modells; der öffentliche Prototyp begrenzt jede erzeugte Welt derzeit auf 60 Sekunden. Man beschreibt seine Umgebung, wählt, ob man geht, reitet, fliegt oder fährt, und spaziert dann hinein. Das ist mehr als Spielerei: Solche generierten Welten dienen zunehmend als Trainingsumgebungen für KI-Agenten, und Waymo hat Anfang 2026 auf Genie-3-Basis ein spezialisiertes Weltmodell für die Simulation autonomen Fahrens aufgebaut.

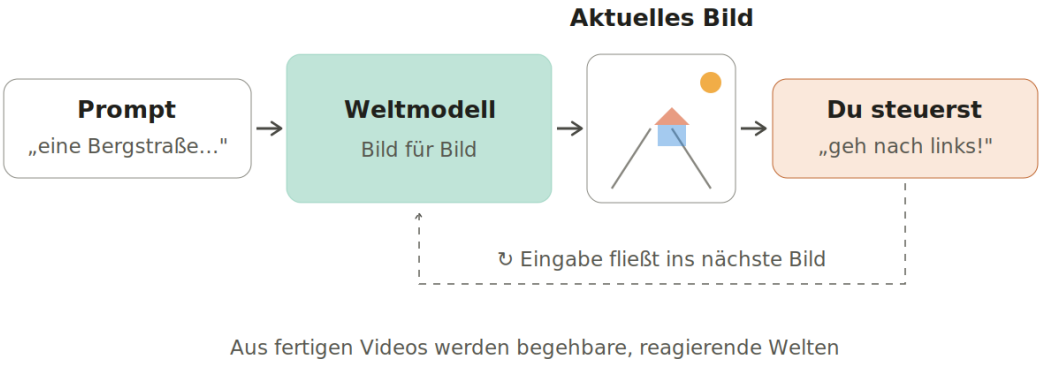

Abbildung 13: Das Weltmodell-Prinzip: Jede Nutzereingabe fließt in das jeweils nächste erzeugte Bild ein – aus Videos werden begehbare Welten.

Was darüber hinaus erwartet wird, folgt aus den drei Baustellen fast zwangsläufig: Erzeugung in Echtzeit statt im Wartemodus, sodass man Ergebnisse während des Entstehens umlenkt; Modelle, die Text, Bild, Ton und Video gemeinsam verstehen und erzeugen; und – als gesellschaftliche Kehrseite der Qualität – Herkunftsnachweise: Weil KI-Inhalte zunehmend nicht mehr vom Echten zu unterscheiden sind, haben Regulierungsbehörden in den USA und der EU inzwischen klarere Vorgaben für KI-erzeugte Inhalte etabliert, und Wasserzeichen sowie Kennzeichnungspflichten dürften so selbstverständlich werden wie heute das Impressum.

Merksatz: Die KI-Ausgabe wird gerade auf drei Wegen besser – *schneller* durch wenige große Sprünge statt vieler Trippelschritte, *kontrollierbarer* durch Steuerung und Konsistenz statt Würfelglück, und *klüger* durch eingebaute Denkzeit mit eigenem Prüfer – und der nächste große Schritt macht aus dem Bild, das man ansieht, eine Welt, die auf einen reagiert.

6. Die Bausteine im Einsatz – physische KI im automatisierten Fahrzeug (SAE L3–L5)

Mit diesem Kapitel verlässt die KI den Bildschirm: Aus erzeugten Bildern werden wahrgenommene Szenen, aus Ausgaben werden Handlungen – der Schritt, den die Branche gerade „Physical AI“ (physische KI) tauft. Dabei greifen das bisher Erläuterte und das automatisierte Fahrzeug (SAE Level 3 bis 5: von der bedingten Automatisierung, bei der der Mensch auf Aufforderung übernimmt, bis zum fahrerlosen Betrieb; die Stufen des US-Ingenieursverbands SAE sind der Branchenstandard) enger ineinander, als es zunächst wirkt: Die Bausteine der Kapitel tauchen dort an vier Stellen wieder auf – als *Arbeitspferd* (der Encoder), als *Wächter* (der Autoencoder-Trick), als *Fächer der Zukünfte* (die Regionen-Idee des VAE) und als *Werkzeugkasten der Absicherung* (die generative Seite). Und die Frage, wo Objekt- und Situationserkennung ansetzen, hat eine präzise Antwort: nicht auf Rohpixeln, sondern **auf dem Code**.

Wo Objekt- und Situationserkennung ansetzen

Die Wahrnehmungskette ist im Kern die erste Hälfte unseres Autoencoders. Das Backbone-Netz moderner Perzeption *ist* ein Encoder: Es presst die Millionen Pixel mehrerer Kameras zu einer kompakten Merkmalsrepräsentation zusammen – heute typischerweise fusioniert in eine gemeinsame Vogelperspektiven-Darstellung, also wörtlich eine *Codekarte der Szene*. Die **Objekterkennung** setzt genau dort an: in Form kleiner Zusatznetze, die den Code auslesen („was ist wo“ – Fahrzeug, Fußgänger, Ampelzustand, Freiraum; semantische Segmentierung („Malen nach Zahlen“, jedes Pixel bekommt eine Bedeutungsklasse) nutzt sogar buchstäblich Encoder-Decoder-Sanduhren). Die **Situationserkennung** setzt eine Ebene höher an: auf der *zeitlichen Folge* dieser Szenen-Codes – Verfolgung über die Zeit (Tracking), Interaktion, Absichtsschätzung – und speist die Planung. Und parallel dazu, bewusst *außerhalb* der Funktionskette, sitzt der Sicherheitsmechanismus: der Wächter.

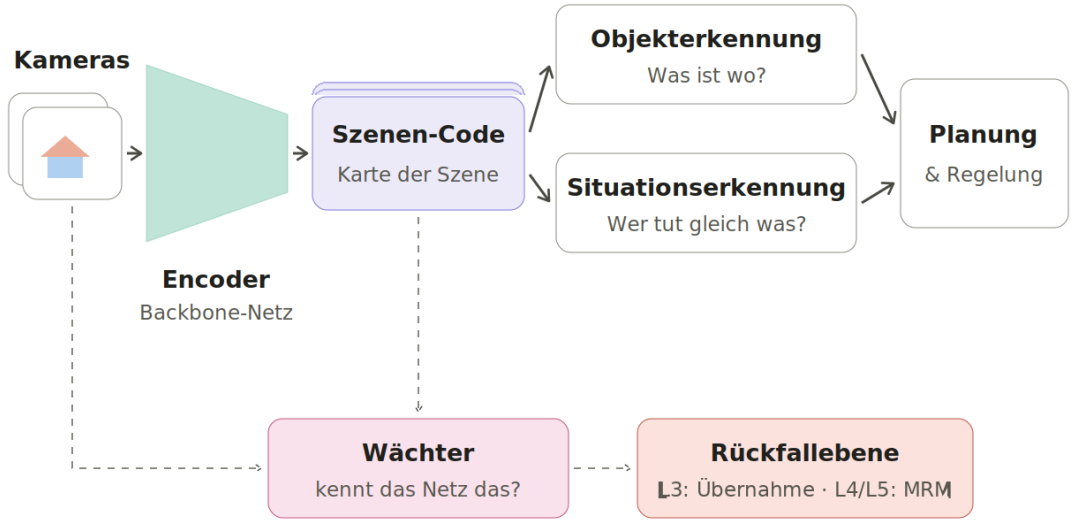

Abbildung 14: Kamerawahrnehmung im automatisierten Fahrzeug: Objekterkennung setzt auf dem Szenen-Code an, Situationserkennung auf dessen zeitlicher Folge – der Wächter überwacht parallel.

Der Wächter: SOTIF und die Normlandschaft

Der Wächter ist der Kapitel-1-Trick in sicherheitskritischer Reinform: ein Autoencoder (oder seine VAE-Variante), der ausschließlich auf die Normalität des vorgesehenen Einsatzbereichs (ODD) trainiert ist und bei hohem Rekonstruktionsfehler anschlägt – so erkennt er zur Laufzeit Eingaben, die das Netz aus dem Training nicht kennt („out-of-distribution"). Normativ ist das exakt die Nahtstelle: „Funktionale Sicherheit" (*ISO 26262 – Functional safety*) adressiert Fehlfunktionen der E/E-Umsetzung (Elektrik und Elektronik: Plattform, Rechner, klassische Software), während die *Insuffizienzen der trainierten Funktion* (übersehene Objekte – „False Negatives" – und Fehlklassifikationen) samt ihrer auslösenden Bedingungen (Triggering Conditions) in „SOTIF" („Sicherheit der Sollfunktion", *ISO 21448 – Safety of the intended functionality*) liegen und „Sicherheit und künstliche Intelligenz" (*ISO/PAS 8800 – Safety and artificial intelligence*) die KI-spezifischen Bausteine dazu liefert: Datenvollständigkeit über den vorgesehenen Einsatzbereich (ODD), Robustheit, Unsicherheitsschätzung, Laufzeitüberwachung, Drift. Der Wächter macht aus *unbekannt-unsicher* ein *erkannt-unsicher* – und damit erst behandelbar. Die Levelabhängigkeit ist dabei der entscheidende Unterschied: Bei L3 mündet der Alarm in die Übernahmeaufforderung an den Fahrer – übernimmt dieser nicht, muss auch das L3-System selbst risikominimal anhalten –, bei L4/L5 muss das System von vornherein selbst in den risikominimalen Zustand – per risikominimalem Manöver (MRM). Das „Ich-kenne-das-nicht-Signal" wird damit selbst zu einer sicherheitsrelevanten Fähigkeit mit eigenen Anforderungen an Verpasser- und Fehlalarmraten (False-Negative-/False-Positive-Raten) – flankiert von klassischen Mechanismen wie diversitärer Redundanz – Plausibilisierung durch verschiedenartige Sensoren, etwa Radar und Lidar (Laser-Abstandsmessung) – um die KI-Blackbox herum.

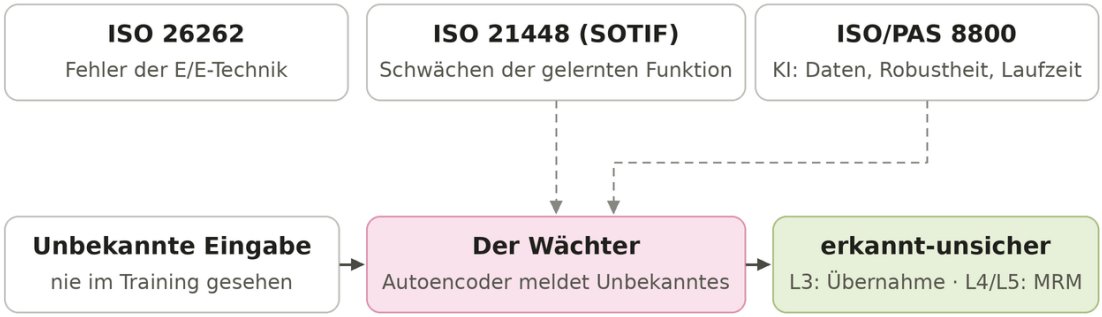

Abbildung 15: Die Normlandschaft und der Wächter: ISO 26262, SOTIF und ISO/PAS 8800 teilen die Zuständigkeiten – der Autoencoder meldet Unbekanntes zur Laufzeit.

Prädiktion ist Erzeugung: Region statt Punkt

Die Situationserkennung führt direkt zur VAE-Pointe der Reihe, denn sie muss *Zukünfte* vorhersagen – und das ist eine generative Aufgabe. Der Fußgänger am Bordstein hat mehrere plausible Zukünfte: queren oder warten. Eine deterministische Ein-Punkt-Prognose mittelt diese Moden – exakt der „weiche Decoder"-Fehler aus Kapitel 2 und 4 – und landet unphysikalisch in der Mitte, wo keine der beiden Zukünfte liegt. Deshalb sind CVAE-basierte und inzwischen diffusionsbasierte Trajektorien-prädiktoren Stand der Technik. Ein CVAE (Conditional VAE) ist ein VAE mit Spickzettel: Der Szenen-kontext geht als Bedingung in Encoder und Decoder ein, und die Zufallsregion trägt nur noch die eine

offene Frage – welche Zukunft? Jeder Punkt der Region steht für genau eine Zukunft; wertet man mehrere aus, entsteht der Fächer gewichteter Zukünfte, mit dem die Planung konservativ rechnen kann. Die diffusionsbasierten Verwandten übertragen dasselbe Prinzip auf das Handwerk aus Kapitel 3: Sie entrauschen, gelenkt vom Szenenkontext, eine zufällige Bahn schrittweise zu einer plausiblen Trajektorie – jeder Durchlauf liefert eine Zukunft, mehrere Durchläufe füllen wieder den Fächer.

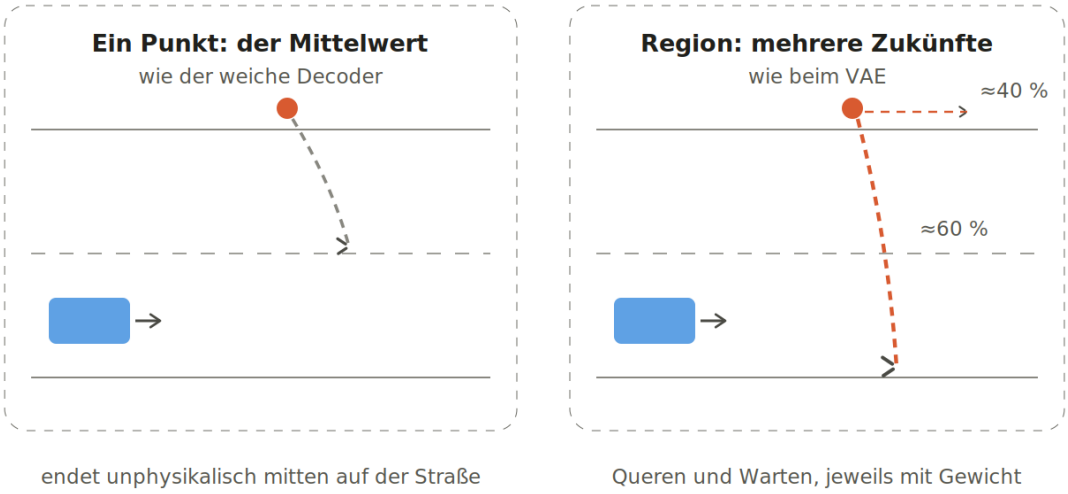

Abbildung 16: Trajektorienprädiktion: Die Ein-Punkt-Prognose mittelt die Zukünfte unphysikalisch – die Regionen-Idee des VAE liefert mehrere gewichtete Zukünfte.

Generative KI und Planung: drei Rollen

Wie steht die generative KI damit insgesamt zur Wahrnehmung als Zulieferin der Planung? Am klarsten wird es entlang von drei Rollen.

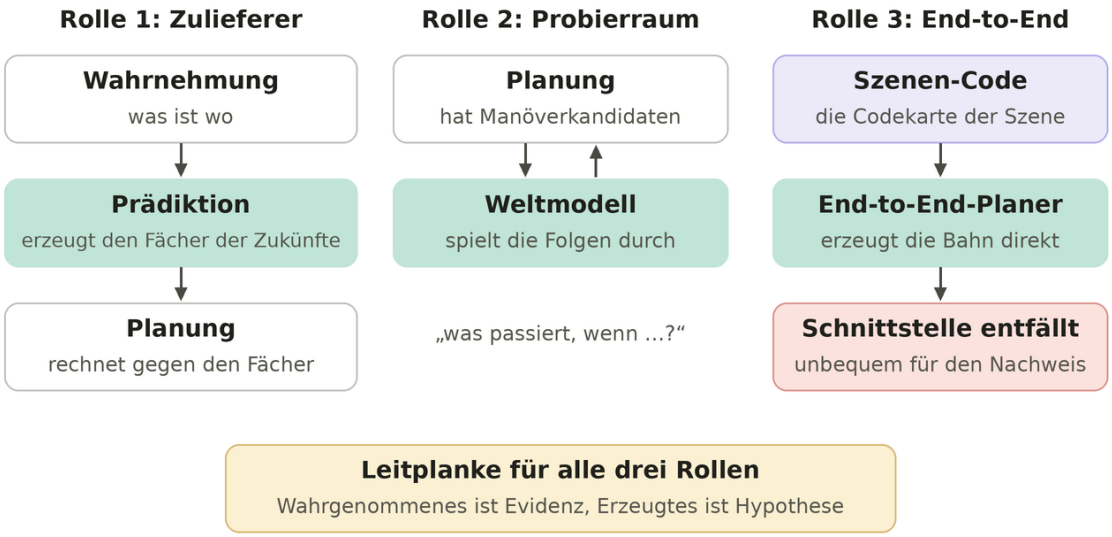

Abbildung 17: Drei Rollen der generativen KI rund um die Planung: Zulieferer des Fächers, Probierraum neben der Planung, End-to-End anstelle der Schnittstelle.

Erstens *in der Zulieferung selbst*: Klassisch liefert die Perzeption der Planung ein Zustandsbild („was ist wo“). Generativ ist daran heute schon das letzte Glied – die Prädiktion. Sie erzeugt den Fächer möglicher Zukünfte, und die Planung erhält damit nicht mehr nur „was ist“, sondern zusätzlich eine erzeugte Verteilung über „was gleich sein könnte“, gegen die sie statt gegen eine Punktschätzung plant.

Zweitens *neben der Planung*: Mit Weltmodellen (Kapitel 5) wird generative KI vom Zulieferer zum Probierraum – die Planung kann Manöverkandidaten intern durchspielen („was passiert, wenn ich jetzt bremse statt auszuweichen?“) und den besten wählen; die Wahrnehmung liefert den Startzustand, das generative Modell die Konsequenzen.

Drittens *anstelle der Schnittstelle*: In End-to-End-Architekturen verschwimmt die saubere Übergabe per Objektliste – die Planung arbeitet direkt auf dem Szenen-Code, teils erzeugen Diffusions-Planer sogar die Ego-Trajektorie (die Bahn des eigenen Fahrzeugs) selbst. Das ist leistungsfähig, für die Absicherung aber die unbequemste Variante, weil die prüfbare Schnittstelle zwischen Wahrnehmen und Entscheiden entfällt.

Für alle drei Rollen gilt dieselbe Leitplanke: **Wahrgenommenes ist Evidenz, Erzeugtes ist Hypothese.** Generative KI darf Zukünfte, Konsequenzen und Szenarien liefern – aber nie den sensorisch belegten Ist-Zustand überschreiben.

Generative Absicherung: aus unbekannt wird bekannt

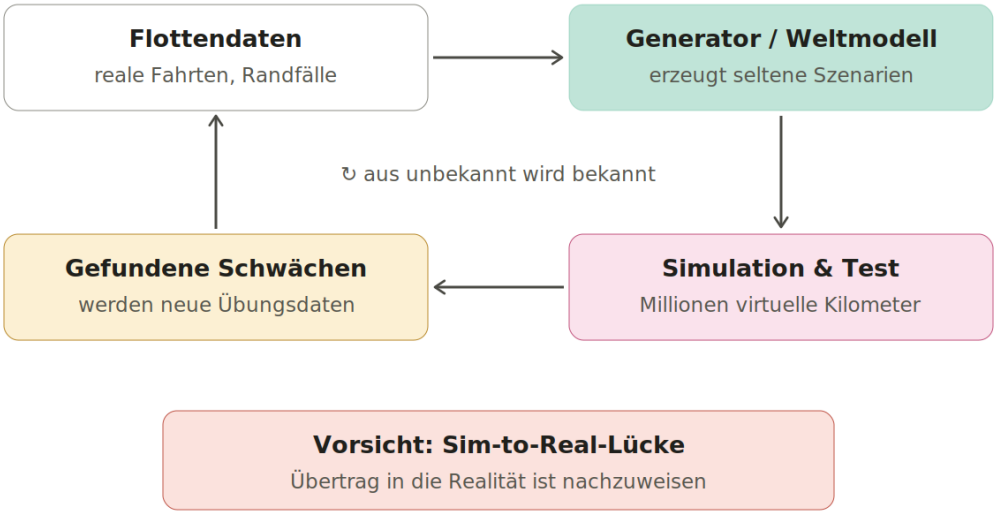

Abbildung 18: Generative KI im Absicherungskreislauf – mit nachweispflichtiger Sim-to-Real-Lücke.

Die Kapitel 3 und 5 liefern schließlich den Werkzeugkasten, mit dem man die SOTIF-Wolke der Unbekannten systematisch verkleinert. Diffusions- und Weltmodelle erzeugen und variieren genau die seltenen Szenarien, an denen Flottendaten arm sind: das Kind hinter dem Ball, Blendung, Starkregen, exotische Objekte und Ladungsverluste. Sie resimulieren aufgezeichnete Fahrten mit „Was-wäre-wenn-Varianten“ und liefern Millionen virtueller Kilometer für szenariobasiertes Testen (methodisch etwa im Rahmen von ISO 34502 – szenariobasierte Sicherheitsbewertung) – exakt die Schiene, auf der Waymo sein Genie-3-basiertes Weltmodell für die Simulation autonomen Fahrens einsetzt. Für den Sicherheitsnachweis gehört allerdings die Kehrseite in den Safety Case (die

dokumentierte Sicherheitsargumentation): Die Sim-to-Real-Lücke – der Abstand zwischen Simulation und Wirklichkeit – ist nachweispflichtig, und generierte Daten erben die Verzerrungen ihres Generators – prüft man ein gelerntes System überwiegend gegen eine *verwandte* gelernte Verteilung, drohen korrelierte blinde Flecken. ISO/PAS 8800 adressiert das über die Anforderungen an Datenqualität und Repräsentativität; die Argumentation, warum synthetische Evidenz zählt, muss explizit geführt werden.

Merksatz – Wahrnehmung und Absicherung: In der L3–L5-Kamerawahrnehmung ist der *Encoder* das Arbeitspferd – die Objekterkennung setzt auf seinem Szenen-Code an, die Situationserkennung auf dessen zeitlicher Folge –, der *Autoencoder* ist der Wächter, der Unbekanntes meldet und damit unbekannt-unsicher in erkannt-unsicher überführt, die *Regionen-Idee des VAE* ist der Schlüssel zur ehrlichen, multimodalen Prädiktion, und die *generative KI* ist der Werkzeugkasten, mit dem aus unbekannt-unsicher Schritt für Schritt bekannt-abgesichert wird.

Hinter der Planung: das Verhalten auf zwei Zeitskalen

Am Ausgang der Planung dreht sich das Blatt gegenüber der Wahrnehmung. Ein Kamerabild lässt sich zur Laufzeit kaum unabhängig auf Richtigkeit prüfen – deshalb brauchte es dort den lernenden Wächter. Eine geplante *Trajektorie* dagegen ist ein konkretes, nachrechenbares Objekt: Positionen, Geschwindigkeiten, Beschleunigungen für die nächsten Sekunden. Hier kann erstmals gegen explizite Regeln geprüft werden – und genau das geschieht, auf zwei Zeitskalen: *im Takt* (verhindern, bevor etwas passiert) und *über die Fahrt hinweg* (erkennen, aufzeichnen, lernen).

Im Takt: vorschlagen und prüfen

Im Takt heißt das Muster **Vorschlagen und Prüfen.** Der komplexe, zunehmend gelernte Planer schlägt eine Bahn vor – aber ein bewusst *einfacher, regelbasierter* Prüfer entscheidet, ob sie gefahren wird.

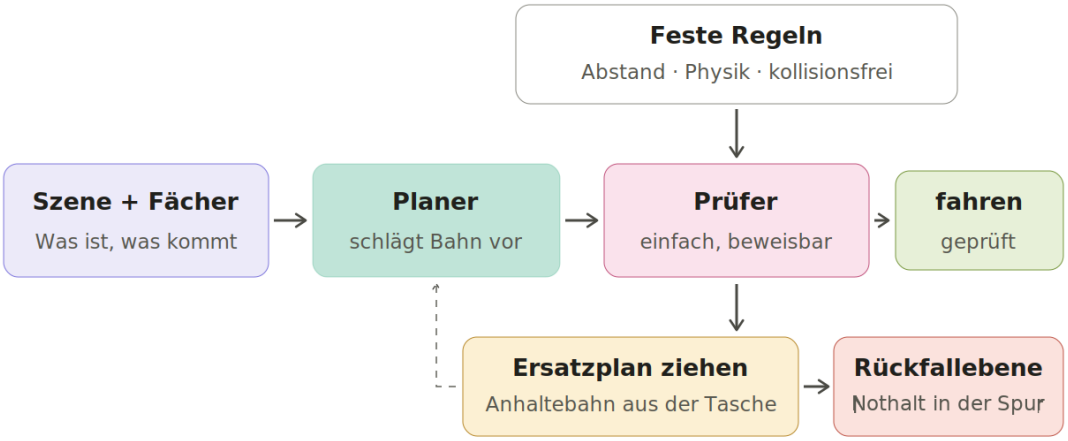

Abbildung 19: Vorschlagen und Prüfen: Der komplexe Planer schlägt eine Bahn vor, der einfache Prüfer entscheidet – mit Anhaltebahn in der Tasche und Rückfallebene.

Geprüft wird gegen den gesamten Prädiktions-Fächer (kollisionsfrei gegenüber *allen* plausiblen Zukünften, nicht nur der wahrscheinlichsten), gegen die Fahrphysik (Reibwert, Querbeschleunigung) und gegen formalisierte Abstandsregeln – Rahmenwerke wie RSS (*Responsibility-Sensitive Safety*, sinngemäß: die Sorgfaltspflichten des Straßenverkehrs, in Formeln gegossen) machen aus „zu nah“ und „zu spät gebremst“ exakte Formeln mit Mindestabständen und vorgeschriebener Reaktion. Typische Unzulänglichkeiten werden damit zu prüfbaren Bedingungen, *bevor* sie auf der Straße landen. Fällt die Prüfung durch, greift die Kaskade: Der Planer muss nachbessern, und für den Ernstfall liegt in jedem Zyklus eine vorab geprüfte **Anhaltebahn** in der Tasche – das vorbereitete risikominimale Manöver; reicht selbst die nicht mehr, bleibt als letzte Ebene der Nothalt in der Spur. Bei L3 wird die Übernahmeaufforderung bereits beim Erreichen der Systemgrenze ausgelöst und gibt dem Fahrer ein Zeitbudget – übernimmt er nicht oder fehlt die Zeit, führt auch das L3-System das risikominimale Manöver selbst aus. Die Architektur-Pointe für den Sicherheitsnachweis: Der Prüfer ist absichtlich *nicht* gelernt, sondern klassische, beweisbare Software – der schmale, hoch abgesicherte Monitor über dem komplexen Kanal, ganz im Sinne der ISO-26262-Dekomposition. Der Prüfer aus den Kapiteln 3 und 5 hat hier also seinen dritten Auftritt – diesmal als Trajektorien-Wächter.

Über die Fahrt hinweg: messen, aufzeichnen, lernen

Die zweite Zeitskala ist *die Fahrt selbst*. Hier läuft eine kontinuierliche **Fahrgüte-Messung** mit: Zeitlücken und Abstände zu anderen, *Time-to-Collision* (die rechnerische Restzeit bis zum Zusammenstoß), Häufigkeit harter Brems- und Lenkeingriffe, Spurhaltegüte.

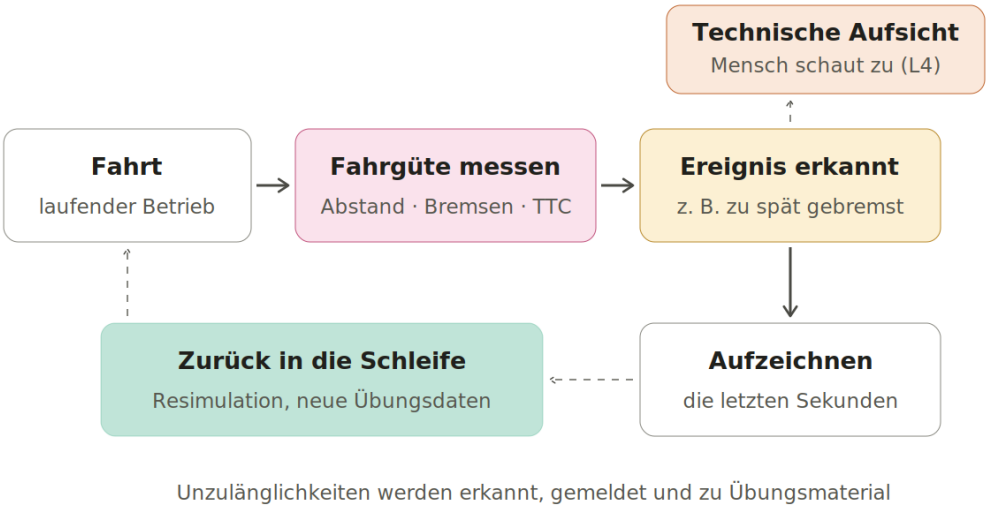

Abbildung 20: Während der Fahrt: Fahrgüte-Metriken lösen Ereignisse aus, die aufgezeichnet, gemeldet und über die Absicherungsschleife zu Übungsmaterial werden.

Überschreitet etwas eine Schwelle – zu spät gebremst, zu dicht aufgefahren –, wird das zum *Ereignis*: Die letzten Sekunden aller Sensor- und Systemdaten werden gesichert – herstellerseitig für die Flottenanalyse; zulassungsseitig ist zumindest das Statusprotokoll Pflicht (Ereignisspeicher DSSAD (*Data Storage System for Automated Driving*): wer fuhr wann – System oder Mensch). Je nach Schwere reagiert das System zudem abgestuft – Tempo raus, Einsatzbereich einschränken, im Zweifel das risikominimale Manöver. Bei L4 in Deutschland kommt eine menschliche Laufzeitinstanz hinzu, die es in dieser gesetzlich verankerten Form sonst nirgends gibt: die **Technische Aufsicht** nach § 1f StVG (Straßenverkehrsgesetz), die zugeschaltet wird, Manöver freigeben und das System deaktivieren kann. Und damit schließt sich der Kreis zum vorigen Abschnitt: Jedes aufgezeichnete Ereignis ist Rohstoff für die Absicherungsschleife aus Abbildung 18 – Resimulation („wäre es ohne Eingriff gutgegangen?“), Ursachenanalyse, neue Trainings- und Testdaten. Die Feldbeobachtung, die SOTIF und AFGBV (Autonome-Fahrzeuge-Genehmigungs-und-Betriebs-Verordnung) ohnehin fordern, wird so vom Pflichtprogramm zum Motor der Verbesserung.

Merksatz – Verhalten: Das Verhalten des Fahrzeugs wird auf zwei Zeitskalen bearbeitet: Im Takt prüft ein einfacher, beweisbarer Prüfer jede vorgeschlagene Bahn gegen Fächer, Fahrphysik und formale Abstandsregeln – mit einer Anhaltebahn in der Tasche –, und über die Fahrt hinweg werden Unzulänglichkeiten gemessen, aufgezeichnet und über die Absicherungsschleife zu Übungsmaterial: verhindern im Moment, lernen über die Flotte.

7. Rund um die Fahrfunktion: der Transformer fährt mit, die Flotte lernt (Stand: Juli 2026)

Kapitel 6 hat die Fahrfunktion abgeschritten – wahrnehmen, vorhersagen, planen, verhalten – und ihre Absicherung gleich mit. Damit ist die Geschichte der KI im automatisierten Fahrzeug aber noch nicht auserzählt. Drei Schauplätze fehlen: das Fahrzeug selbst, in das gerade ein neues Familienmitglied einzieht; die Datenfabrik dahinter, in der die Flotte lernt; und der Entwicklungsprozess, in dem die KI den Ingenieuren zur Hand geht. Die gute Nachricht für alle, die von vorn gelesen haben: Fast überall arbeiten auch hier nur die bekannten Bausteine – in neuen Rollen. Wirklich neu ist ein einziger Auftritt, ausgerechnet der des Verwandten, den Kapitel 3 mit einem Satz beiseitegelegt hatte: des Transformers.

Der Transformer fährt mit: Sprachmodelle am Steuer

In Kapitel 3 blieb der Transformer – das „T“ in ChatGPT – am Wegesrand stehen: Text-KI geht einen anderen Weg. Inzwischen ist dieser Weg ins Fahrzeug eingebogen. **Vision-Language-Action-Modelle** (VLA; sinngemäß: ein Modell, das sieht, in Sprache denkt und daraus Fahrhandlungen ableitet) verbinden die Wahrnehmung aus Kapitel 6 mit dem Weltwissen und der Denkzeit aus Kapitel 5. Der Gewinn zeigt sich genau dort, wo rein aus Fahrszenen gelernte Systeme traditionell schwächeln: am langen Ende der seltenen Fälle. Ein Netz, das nur Verkehr gesehen hat, kennt das Sofa auf der Fahrbahn nicht; ein Modell, das zusätzlich mit dem Wissen der Welt trainiert wurde, kann sich einen Reim darauf machen – ebenso auf das Handzeichen des Polizisten oder die ausgefallene Ampel an der Kreuzung.

Der zweite Gewinn ist für die Absicherung mindestens so interessant: Diese Modelle denken in nachlesbaren Zwischenschritten. NVIDIA hat Anfang 2026 mit Alpamayo 1 ein offenes 10-Milliarden-Parameter-Modell vorgestellt, das aus Kamerabildern nicht nur eine Trajektorie erzeugt, sondern die **Begründungsspur** gleich mitliefert – eine Kette von Zwischenschritten, warum genau diese Bahn gewählt wurde. Seit dem ersten Quartal 2026 fährt der Ansatz in Serie im Mercedes-Benz CLA, für Robotaxis steht mit Alpamayo 2 Super ein 32-Milliarden-Parameter-Modell bereit, und Li Auto hat im März 2026 sein eigenes Fahr-Grundmodell MindVLA gezeigt. Die Denkschleife aus Kapitel 5 sitzt damit wörtlich mit im Auto.

Für den Sicherheitsnachweis ist das janusköpfig. Die Begründungsspur ist ein neuartiges Nachweisartefakt – erstmals kann man einem gelernten Planer beim Denken zusehen und seine Entscheidung im Ereignisfall nachlesen. Zugleich erbt das Fahrzeug die bekannten Schwächen der Sprachmodelle: Eine flüssige Begründung kann falsch sein (*Halluzination*), Denkzeit kostet Reaktionszeit, und ob die Begründung wirklich die Ursache der Handlung ist oder nur ihre schöne Nacherzählung, ist eine offene Forschungsfrage. Die Architektur-Antwort aus Kapitel 6 gilt deshalb unverändert: Auch das klügste VLA-Modell ist der komplexe Kanal – und über ihm wacht der einfache, beweisbare Prüfer.

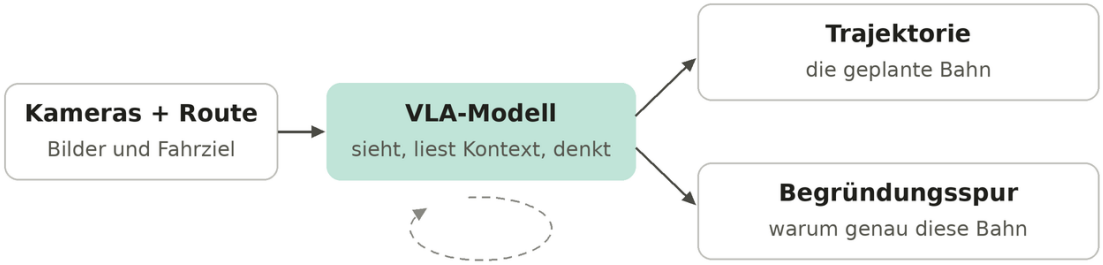

Abbildung 21: Das VLA-Modell im Fahrzeug: Aus Bildern und Kontext entstehen Trajektorie und Begründungsspur – die Denkzeit aus Kapitel 5 fährt mit.

Wie eng das Sprachmodell eingekoppelt wird, ist dabei selbst eine Architekturentscheidung – und der Markt zeigt Anfang 2026 zwei Antworten. Dem fahrenden VLA-Modell stellt Mobileye mit **VLSA** (Vision-Language-Semantic-Action; sinngemäß: das Modell liefert Bedeutung statt Bahn) eine beratende Variante gegenüber: Ein langsam denkendes Sprachmodell gibt wie ein begleitender Fahrlehrer nur strukturierte semantische Hinweise – „Vorsicht, der Polizist regelt den Verkehr“ –, während Trajektorie und sicherheitskritische Regelung im schnellen, formal abgesicherten Fahr-Stack (dem geschichteten Softwarepaket aus Wahrnehmen, Planen, Regeln) verbleiben; das generative Modell steht damit gar nicht erst in der Sicherheitsschleife. Der Preis ist eine neue, sorgfältig zu spezifizierende Schnittstelle: Der Rat muss richtig, rechtzeitig und richtig verstanden sein – und der schnelle Pfad muss auch ohne ihn sicher bleiben.

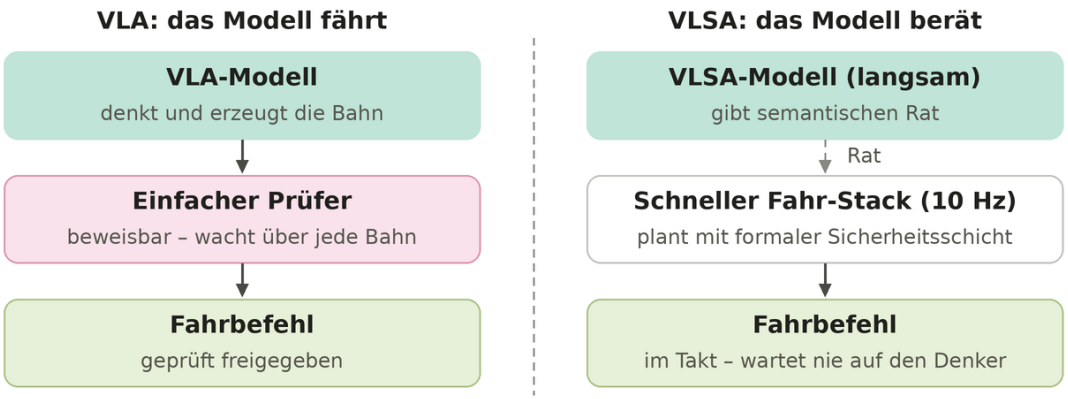

Abbildung 22: Zwei Kopplungen des Transformers: Das VLA-Modell erzeugt die Bahn selbst – der Prüfer wacht; das VLSA-Modell berät nur, die Bahn entsteht im schnellen, formal abgesicherten Stack.

Drei stille Helfer an Bord

Neben dem prominenten Neuzugang arbeiten im Fahrzeug drei unauffälligere KI-Rollen – alle drei mit Bausteinen aus den ersten Kapiteln. Erstens der Blick nach innen: Bei L3 hängt die gesamte Rückfalllogik an der Übernahmeaufforderung, und die setzt voraus, dass laufend geprüft wird, ob der Fahrer überhaupt übernehmen kann – die einschlägige Regelung für solche Systeme (UN R157) verlangt eine Verfügbarkeitserkennung ausdrücklich. Genau das leistet eine Innenraumkamera mit gelernter Blick- und Müdigkeitserkennung: Wahrnehmung wie in Kapitel 6, nur um 180 Grad gedreht. Zweitens der Wächter im Bordnetz: Derselbe Kapitel-1-Trick, der unbekannte Kamerabilder meldet, erkennt auch unbekannten Datenverkehr auf den Datenbussen des Fahrzeugs, vom klassischen CAN-Bus bis zum Ethernet-Bordnetz – als Einbruchserkennung (Intrusion Detection) im Sinne der Cybersecurity-Regelwerke (UN R155 und *ISO/SAE 21434 – Road vehicles, Cybersecurity engineering*). Ein Angriff ist aus Sicht des Autoencoders schlicht eine Anomalie. Drittens der Entrauscher im Wortsinn: Das Handwerk aus den Kapiteln 3 und 4 rechnet Regen, Gischt und Schneetreiben aus Kamerabildern und Lidar-Punktwolken heraus, bevor die Wahrnehmung sie zu sehen bekommt – und die Anomalieerkennung meldet nebenbei, wenn ein Sensor verdreckt oder dekalibriert ist.

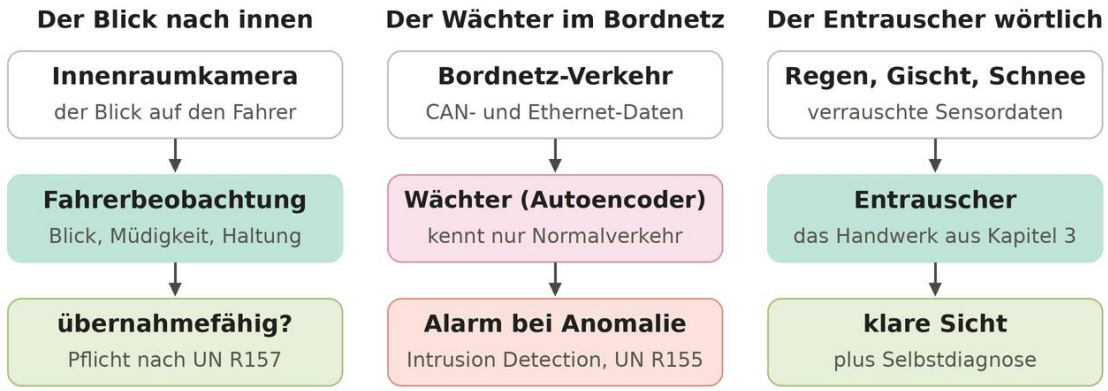

Abbildung 23: Drei stille Helfer an Bord: Fahrerbeobachtung, Bordnetz-Wächter und wörtliches Entrauschen – bekannte Bausteine in Nebenrollen.

Die Datenfabrik: Lehrer, Schüler und der Wächter als Kurator

Kapitel 6 endete mit der Absicherungsschleife: Ereignisse werden zu Übungsmaterial. Die Trainingsseite dieser Schleife ist eine eigene Fabrik, und in ihr regiert das Lehrer-Schüler-Prinzip aus den Kapiteln 4 und 5 im Industriemaßstab. Am Anfang steht ein Auswahlproblem: Millionen Flottenkilometer sind fast vollständig ereignislos. Der Wächter-Trick löst es – ein Neuheitsdetektor behält genau die Szenen, die das System noch nicht kennt; alles andere wäre Beschriftungsaufwand ohne Lerneffekt. Dann übernimmt der Lehrer: Ein großes Modell außerhalb des Fahrzeugs (offboard), das nicht in Echtzeit laufen muss, beschriftet die ausgewählten Szenen automatisch (**Auto-Labeling**) – die Musterlösungen, die früher Heerscharen menschlicher Beschrifter lieferten, erzeugt heute ein Modell für das nächste. Zum Schluss die *Destillation* aus Kapitel 5: Der große Lehrer wird auf einen kleinen, schnellen Schüler eingedampft, der ins Steuergerät passt. NVIDIA nennt für sein offenes Fahr-

Grundmodell genau diese Kette – Feinabstimmung, Destillation, Auto-Labeling – als vorgesehene Verwendung.

Zwei Ergänzungen runden die Fabrik ab. Die Grundausbildung der Wahrnehmungsnetze läuft heute standardmäßig ohne jede Beschriftung – mit **Masked Autoencoders** (sinngemäß: Autoencoder, die absichtlich abgedeckte Bildteile ergänzen): der Kapitel-1-Apparat in neuer Spielart, Lücken füllen statt Rauschen entfernen. Und die Fahrpolitik selbst wird zunehmend per **Reinforcement Learning** nachgeschult (bestärkendes Lernen: Belohnung statt Musterlösung) – und zwar geschlossen in der Simulation. Der Weltmodell-Probierraum aus Kapitel 6 wird damit vom Test- zum Trainingsraum, in dem das System Millionen Situationen durchspielt, für gute Manöver belohnt wird und schlechte folgenlos verlernen darf.

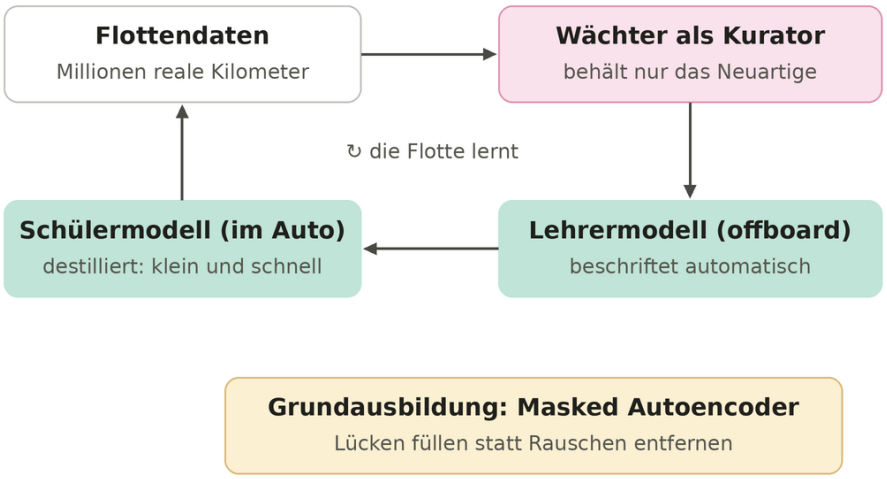

Abbildung 24: Die Datenfabrik: Der Wächter kuratiert die Flottendaten, der Lehrer beschriftet automatisch, die Destillation liefert den Schüler fürs Fahrzeug – und die Grundausbildung übernimmt der Masked Autoencoder.

KI im Entwicklungsprozess: Assistenz, nicht Nachweis

Der dritte Schauplatz liegt vor dem ersten gefahrenen Meter: in der Entwicklung selbst. Sprachmodelle entwerfen Testszenarien aus einer Beschreibung in Alltagssprache („ein Kind tritt bei tief stehender Sonne zwischen parkenden Autos hervor“) und übersetzen sie in formale Szenariosprachen wie OpenSCENARIO; sie liefern Rohfassungen für Gefährdungsanalysen, prüfen Anforderungslisten auf Lücken und Widersprüche und fassen Testkampagnen zusammen. Das ist echte Entlastung – und braucht dieselbe Leitplanke, die Kapitel 6 für das Fahrzeug gezogen hat: *Erzeugtes ist Hypothese*. Ein KI-Entwurf wird erst durch Prüfung und Freigabe eines Menschen zum Nachweis; die Verantwortung – und im Sinne der ISO 26262 auch das Vertrauen in die Werkzeugkette – bleibt beim Ingenieur. Der Prüfer aus den Kapiteln 3, 5 und 6 hat damit seinen vierten Auftritt, und es ist der schönste: Diesmal ist er ein Mensch.

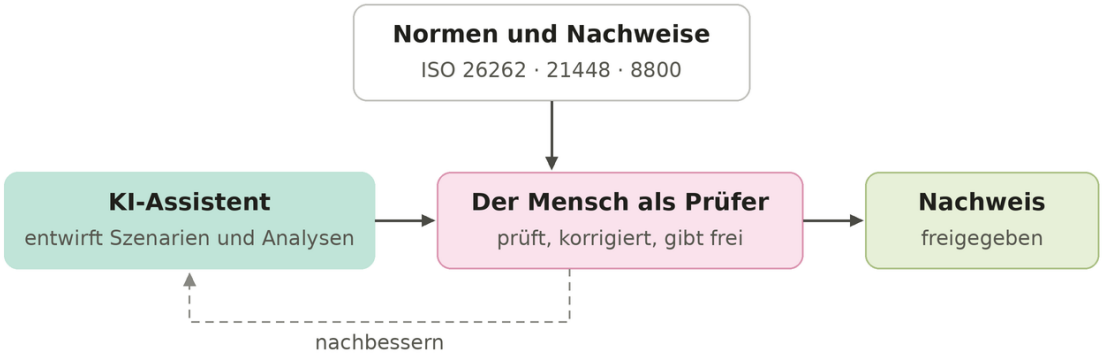

Abbildung 25: Assistenz, nicht Nachweis: Die KI entwirft, der Mensch prüft, korrigiert und verantwortet – der vierte Auftritt des Prüfers.

Merksatz: Rund um die Fahrkette arbeitet dieselbe Bausteinfamilie weiter: Der Transformer zieht als denkendes, sich erklärendes Modell ins Fahrzeug ein – fahrend als VLA oder beratend als VLSA; der Wächter überwacht nebenbei Bordnetz und Sensorik, während eine nach innen gedrehte Wahrnehmung den Fahrer im Blick behält; in der Datenfabrik kuratiert der Wächter die Flottendaten, der Lehrer beschriftet die Musterlösungen, der Schüler wird destilliert – und im Entwicklungsprozess gilt die alte Leitplanke: Die KI entwirft, der Mensch prüft und verantwortet.

Quellen

Die Kapitel 1 bis 4 sowie die methodischen Grundlagen in den Kapiteln 6 und 7 erklären etabliertes Wissen des maschinellen Lernens bzw. der einschlägigen Normen und Regelwerke (ISO 26262, ISO 21448, ISO/PAS 8800, ISO 34502, ISO/SAE 21434, UN R155/R157, StVG/AFGBV).

Die Aussagen zu aktuellen Entwicklungen in den Kapiteln 5 bis 7 (u. a. Waymo/Genie 3, NVIDIA Alpamayo) stützen sich auf folgende, im Juli 2026 abgerufene Quellen:

Google / Google DeepMind: „Project Genie: AI world model now available for Ultra users in U.S.“, blog.google, 29.01.2026.

TechBriefly: „Google launches Project Genie globally for adult AI Ultra users“, techbriefly.com, 20.05.2026.

Wikipedia: „Genie (world model)“, en.wikipedia.org (abgerufen Juli 2026).

Waymo: „The Waymo World Model: A New Frontier for Autonomous Driving Simulation“, waymo.com/blog, 06.02.2026.

WaveSpeed Blog: „Google DeepMind Genie 3: The World Model That Creates Interactive Environments“, 30.01.2026.

Forbes (A. Husain): „Why World Models Like Google’s Project Genie Work“, 29.01.2026.

Renderforest: „AI video generation trends in 2026: what’s actually changing“, Mai 2026.

digen.ai: „AI Video Generation Market Trends: 2026 Industry Forecast“ und „AI Video Generator Trends 2026“, Mai 2026.

X. Zhou (Medium): „The State of AI Video Generation in 2026: 5 Shifts That Actually Matter“, Februar 2026.

tooldirectory.ai: „The state of AI image and video generation (2026)“, Juni 2026.

Introl Blog: „Inference-Time Scaling“, Dezember 2025, sowie Übersichtsarbeiten zur Test-Time-Compute-Skalierung (arXiv, 2024–2026).

NVIDIA Newsroom: „NVIDIA Announces Alpamayo Family of Open-Source AI Models and Tools to Accelerate Safe, Reasoning-Based Autonomous Vehicle Development“, nvidianews.nvidia.com, 05.01.2026.

Electrek: „Nvidia unveils open-source AI for autonomous driving, ships in Mercedes-Benz CLA in Q1 2026“, electrek.co, 05.01.2026.

NVIDIA: „Alpamayo – Open AI for Robotaxis and Autonomous Vehicles“ (u. a. Alpamayo 2 Super, AlpaGym), nvidia.com (abgerufen im Juli 2026).

Li Auto Inc.: „March 2026 Delivery Update“ – Vorstellung des Fahr-Grundmodells MindVLA auf der NVIDIA GTC 2026, 01.04.2026.

Mobileye Blog: „Prof. Amnon Shashua at CES 2026: Robotaxi updates, breakthroughs in AI, and robotics“ (Vision-Language-Semantic-Action, Fast-and-Slow-Architektur), mobileye.com, Januar 2026.

Übersichtsarbeiten zu Vision-Language-Action-Modellen für das automatisierte Fahren (arXiv/CVPR/AAAI, 2025–2026).