

# GANs in der Wissenschaft

Von Johannes Kolhoff

Universität Hamburg

Für das Proseminar Softwareentwicklung in der Wissenschaft

22.06.2020

# Gliederung

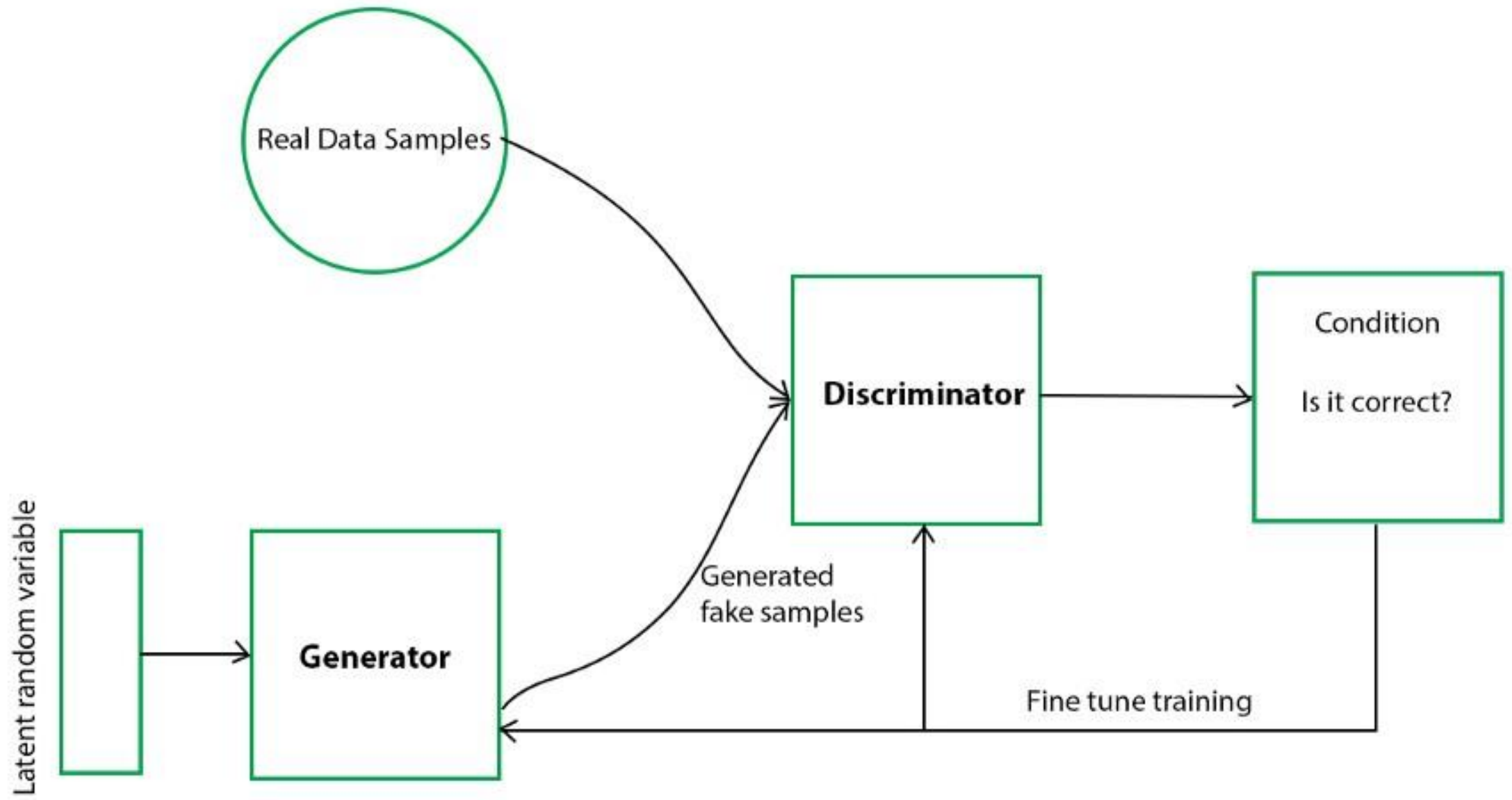
1. Was sind GANs?
2. Wie funktionieren GANs?
3. Beispiel: Verbesserung von Galaxienbildern
4. Beispiel: Datenverarbeitung vom LHC am CERN
5. Bewertung von GANs als wissenschaftliche Methode
6. Zusammenfassung

# Was sind GANs?

- Zuerst 2014 entwickelt
- GAN steht für: Generative Adversarial Networks
- (deutsch: Erschaffende-Gegenspieler Netzwerke)
- Werden genutzt, um zu einem vorgegebenen Datensatz ähnliche neue Daten zu erschaffen
- Zwei Netzwerke optimieren sich durch Antagonismus
- Unüberwachtes Lernen (Unsuperwised Learning )

# Wie funktionieren GANs? | Gliederung

1. Grundkonzept
2. Fehlerfunktion
3. Der Datensatz
4. Der Input
5. Beispiele





D: Detective



R: Real Data



G: Generator (Forger)



I: Input for Generator

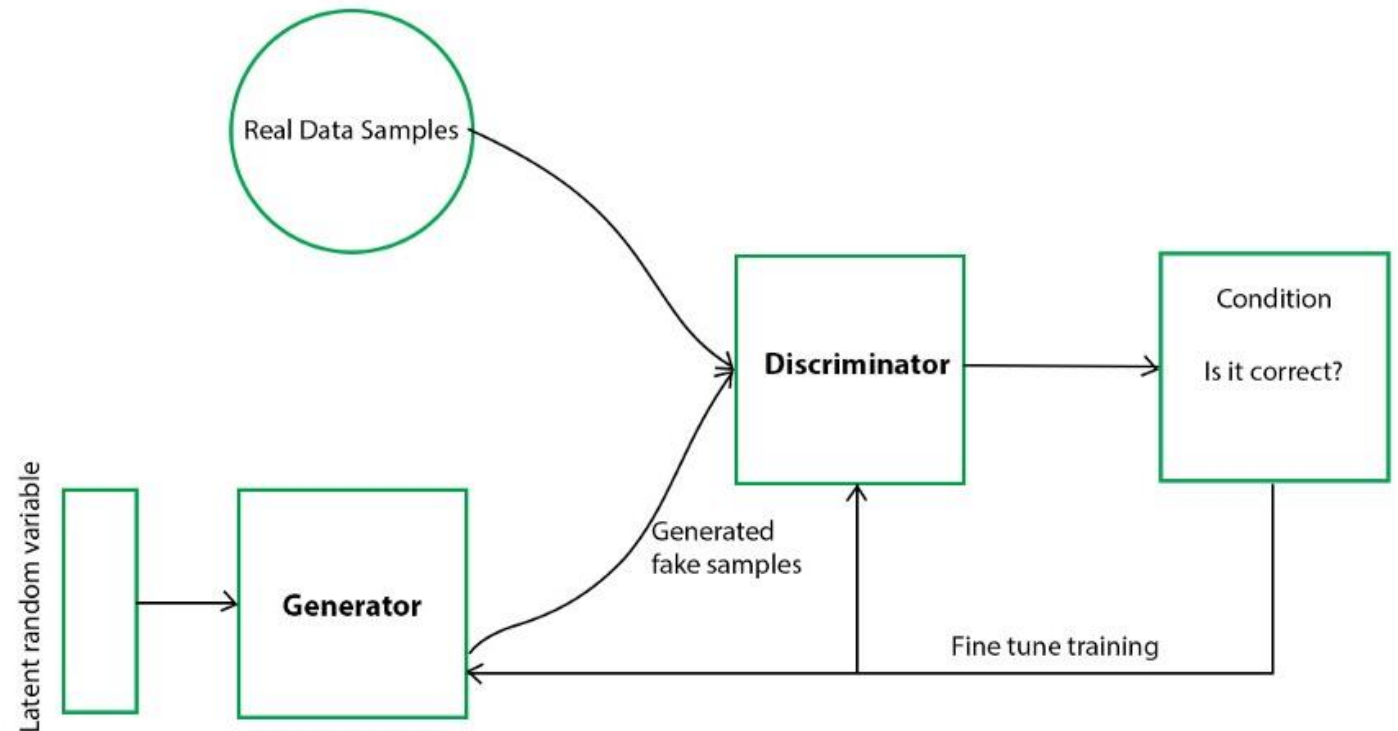


# Bewertung: Die Fehlerfunktion I

- Der Discriminator berechnet eine Fehlerfunktion
- Der Generator (G) bekommt die Fehlerfunktion als Feedback
  - => verbessert Qualität
- Discriminator (D) bekommt "Auflösung"
  - => verbessert Entscheidungen

Am Ende des Trainings:

- D kann Daten ähnlich zum Training erkennen
- G kann Daten ähnlich zum Trainingsset erschaffen

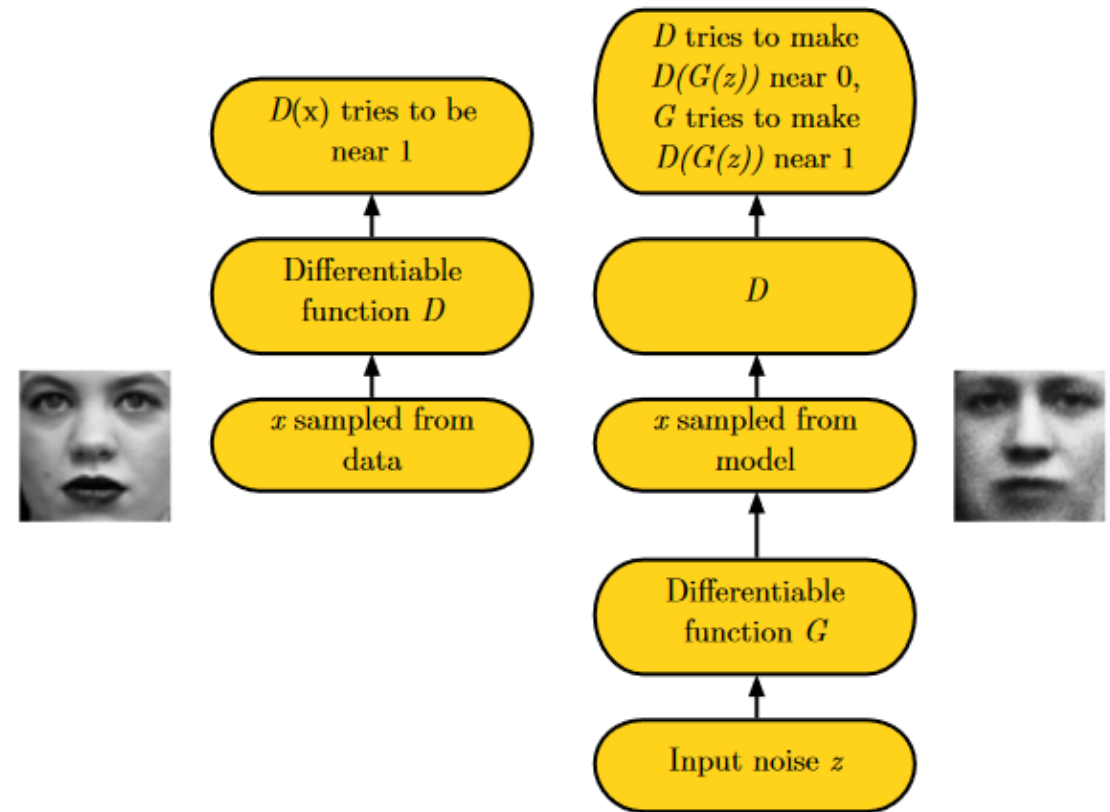


# Bewertung: Die Fehlerfunktion II

$x$  ist ein Element des Datensatzes

$z$  ist der noise

- Generator berechnet  $G(z)$ , eine neue Datei
- Der Discriminator berechnet die  $D(G(z))$  bzw.  $D(x)$
- Der Discriminator will immer für  $x$  1 und für  $G(z)$  0 als Wahrscheinlichkeiten ausgeben
- $D(G(z)) = 0 \mid D(x) = 1$
- Der Generator will die  $D(G(z))$  maximieren
- Der Discriminator will  $D(G(z))$  minimieren
- Wenn ein Gleichgewicht von 0,5 erreicht ist, ist man fertig





# Der Datensatz

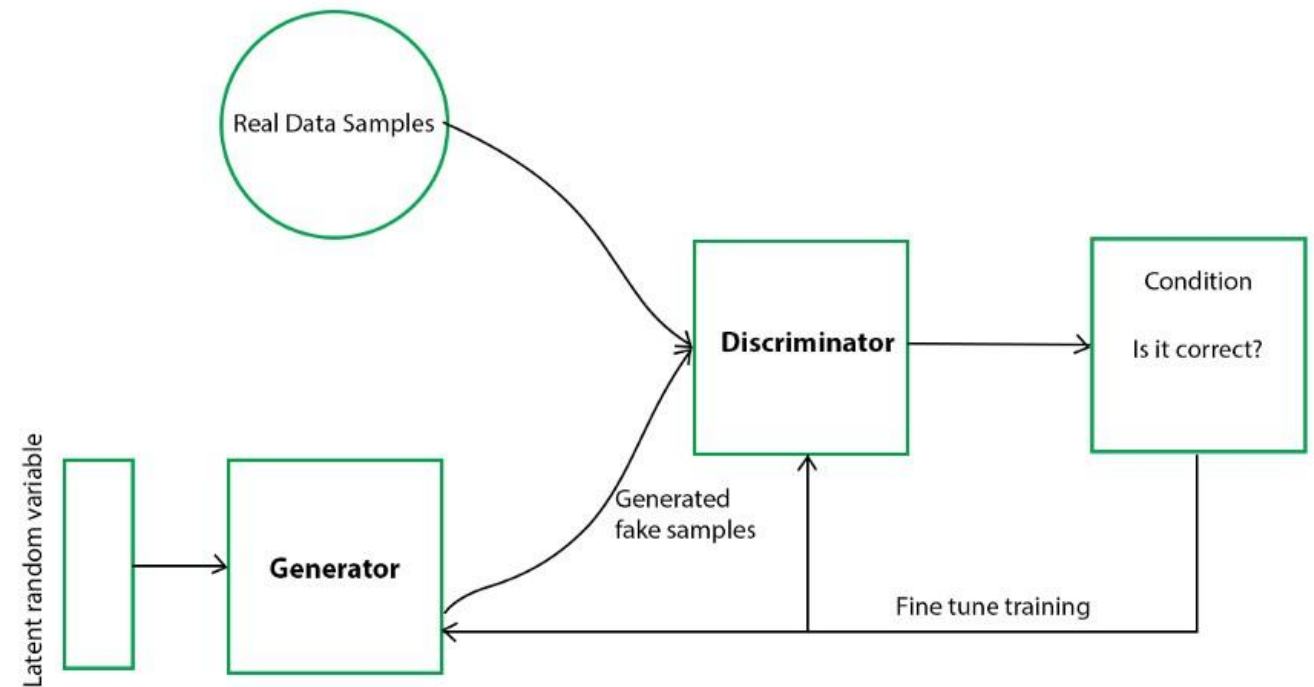
- Die Daten sollten sorgfältig ausgewählt werden  
-> bestimmen maßgeblich über die Qualität der Resultate

Qualität (z.B. Auflösung):

- Gegebenenfalls reduzieren
- Bestimmt die Dauer des Lernprozesses

Anzahl:

- Je größer desto besser
- Aber: je 'gleichförmiger' der Datensatz ist, desto besser das Ergebnis
- Limitierung auf Sonderfälle sinnvoll (z.B. nur Gesichter mit ähnlichem Ausschnitt anstatt Bilder von Menschen in allen möglichen Situationen)



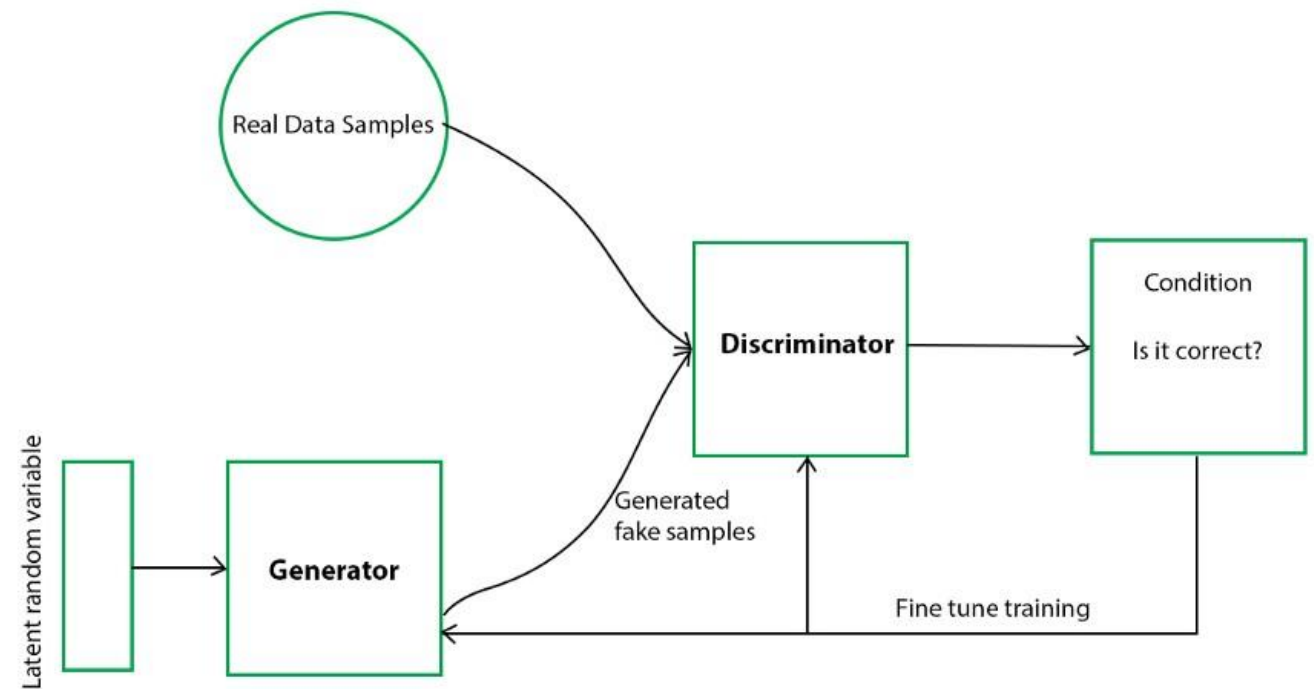
# Der Input

Dem Generator wird ein Input gegeben, aus dem er etwas generiert

- Meistens random noise
- Oder Variablen bzw. Vektoren

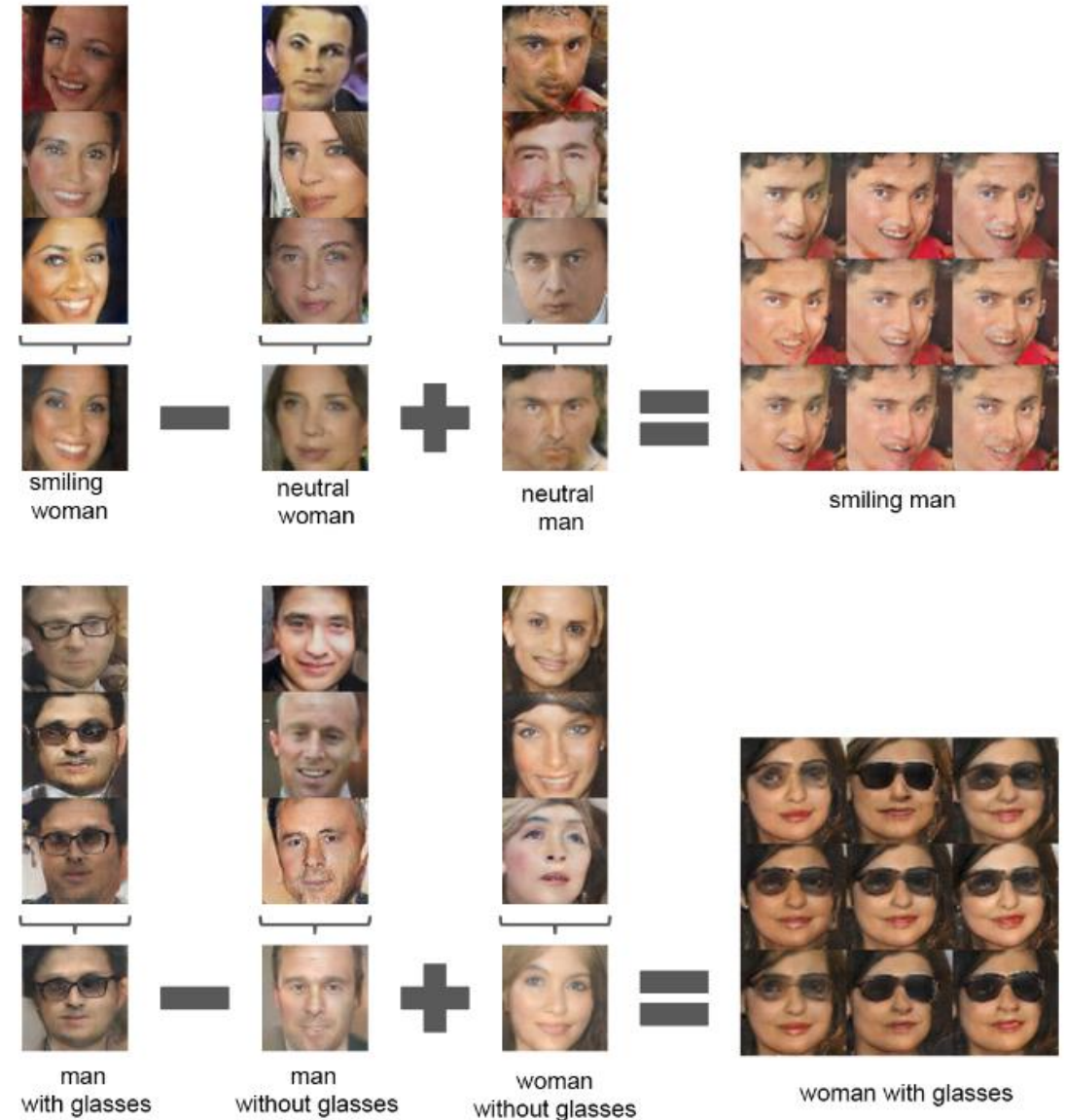
Das hat den Vorteil, dass mit den Eingangsvektoren Attribute des Ergebnisses gesteuert werden können

- Kann aber auch ein beliebiger anderer Datensatz sein
- Bei wissenschaftlichen Anwendungen häufig unkompletter Datensatz



# Beispiel

- Paper aus 2016
- Generierte Bilder in Klassen sortiert
- Es wurde aus den Spalten der Durchschnittsvektor errechnet
- Diese können voneinander abgezogen oder addiert werden
- Zusätzlich wurde noch leichter noise hinzugefügt (Mitte ist ohne noise)



# Progression



2014



2015



2016



2017

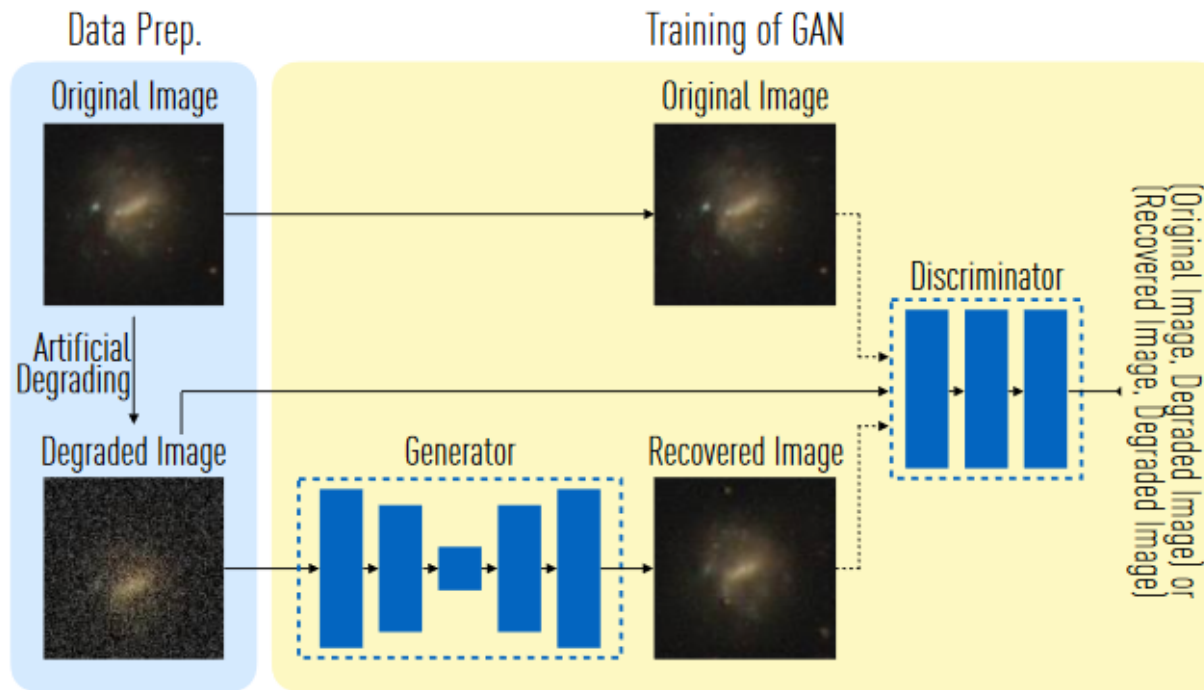
Quelle:

<https://arxiv.org/ftp/arxiv/papers/1802/1802.07228.pdf>

# Beispiel 1: Galaxienbilder

- Problem: Galaxienbilder sollen geschärft werden
- Generator erhält als Input das unscharfe/verpixelte Bild
- Dabei sollen kosmische Objekte erkannt und berücksichtigt werden
- Das Ziel ist eine bessere Bildverbesserungssoftware
- Optimierung auf Galaxienbilder

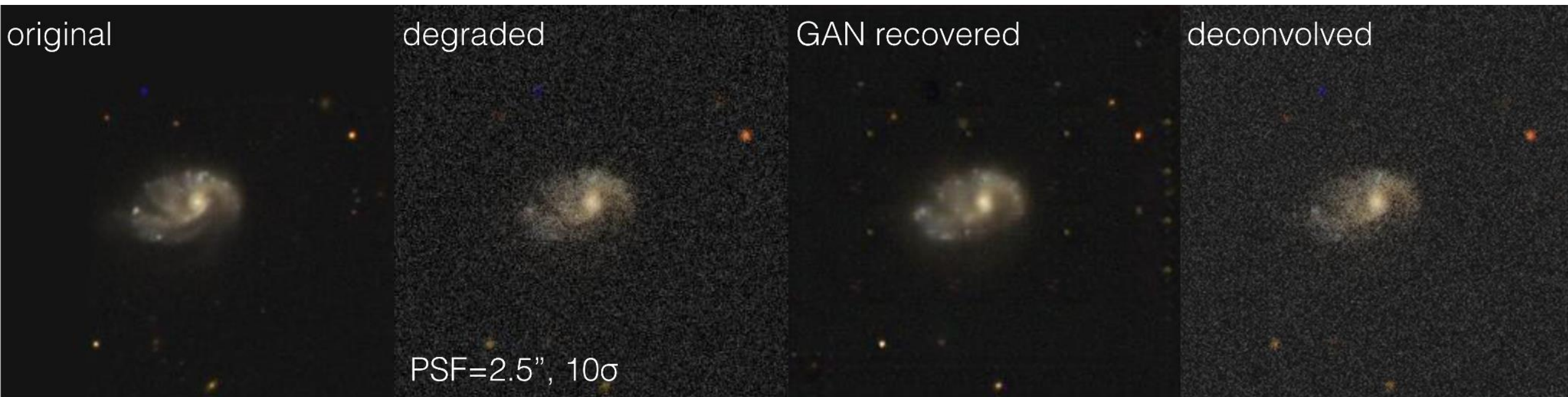
# Trainieren des GANs

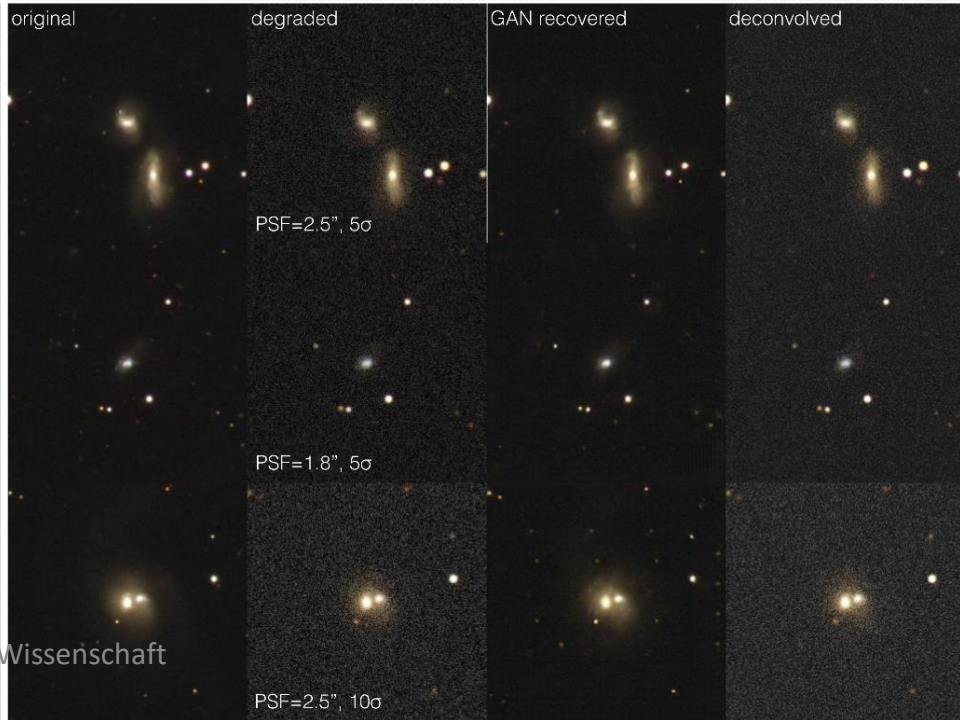
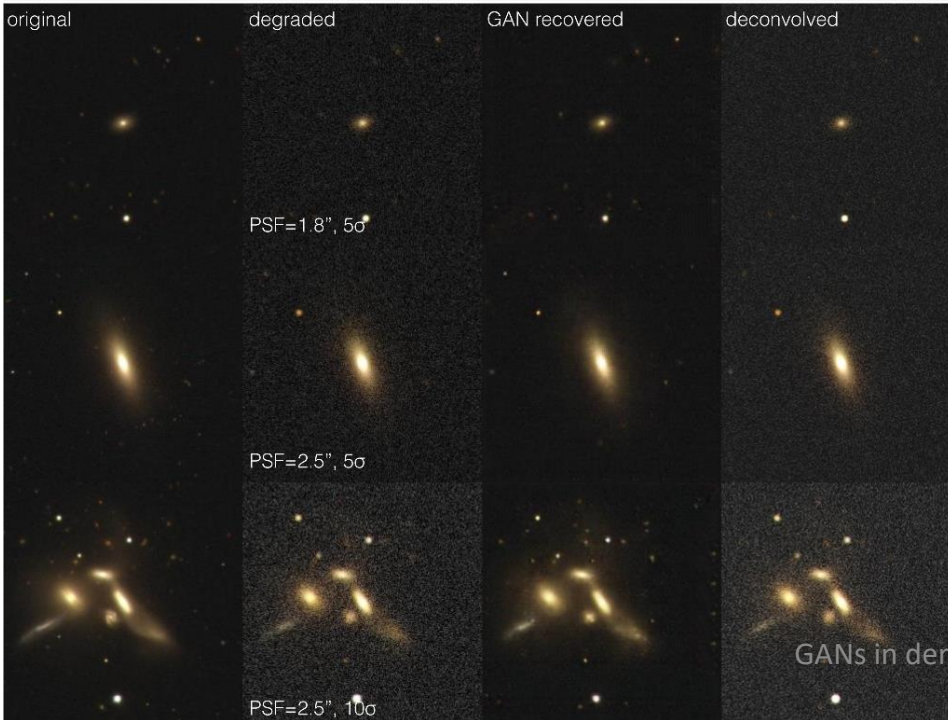
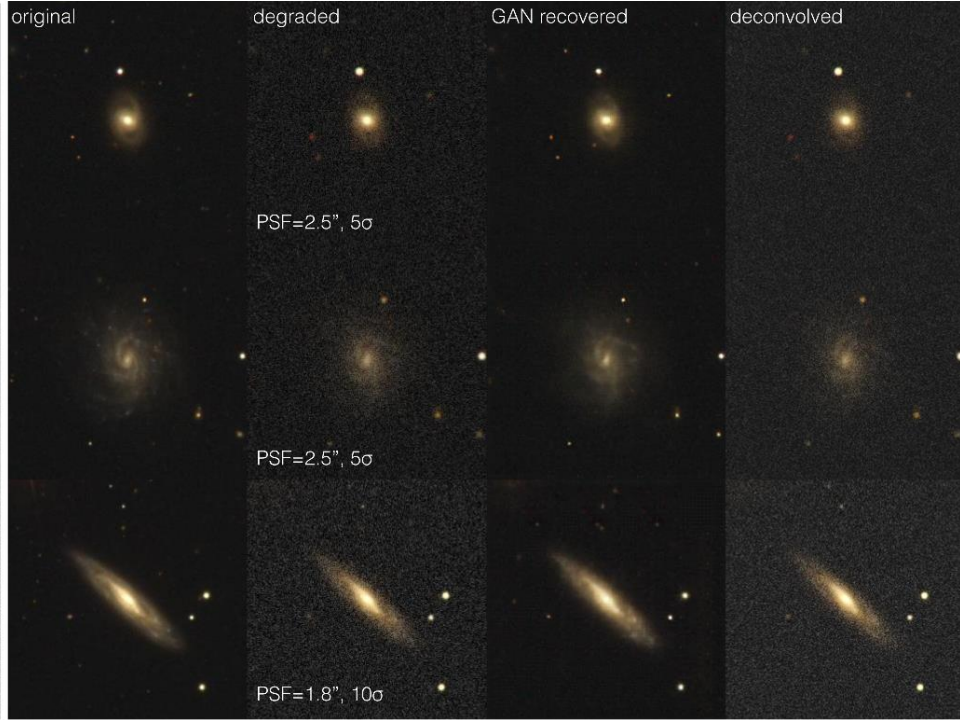
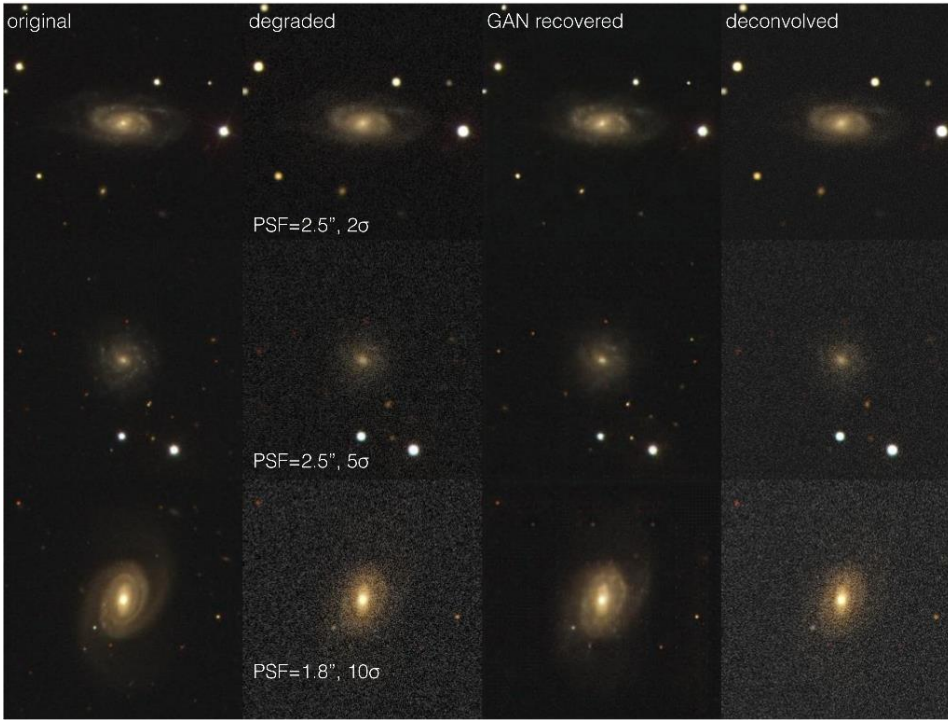


- Hochauflösende Galaxienbilder werden runterskaliert
- Der Generator bekommt als Input die 'schlechte' Version
- Der Discriminator bekommt das runterskalierte Bild, das Original und das vom Generator erzeugte
- Der Discriminator weiß, welches die runterskalierte Version ist und soll entscheiden, was das Original ist



# Ergebnisse







# Zusammenfassung

- Wiederherstellung mit GANs funktioniert deutlich besser als konventionelle Methoden

Aber:

- Selten vorkommende Objekte werden nicht erkannt
- Es wird immer etwas generiert, auch wenn eigentlich nichts zu erkennen ist (falsche Objektivität?)
- Wenn im Trainingsset nur Galaxien mit ähnlicher Rotverschiebung (= Entfernung) sind, funktioniert auch nur (gut) für andere Galaxien in diesem Bereich
- Keine echte Erweiterung des Horizonts
- Aber man könnte Fotos von schlechteren Teleskopen aussehen lassen wie die von besseren

# Beispiel 2: Teilchenberechnung

- Teilchenbeschleuniger LHC am CERN
- Messdaten lassen auf Teilchenflugbahnen schließen
- Kollision -> Spuren werden von Detektoren erfasst -> daraus Flugbahn berechnen
- Problem: Berechnungen sind kompliziert und brauchen viel Zeit
- LHC Sensoren werden verbessert
- 4 fache Datenmenge zu verarbeiten
- Bisherige Methode nicht mehr praktisch (auf aktueller Hardware)

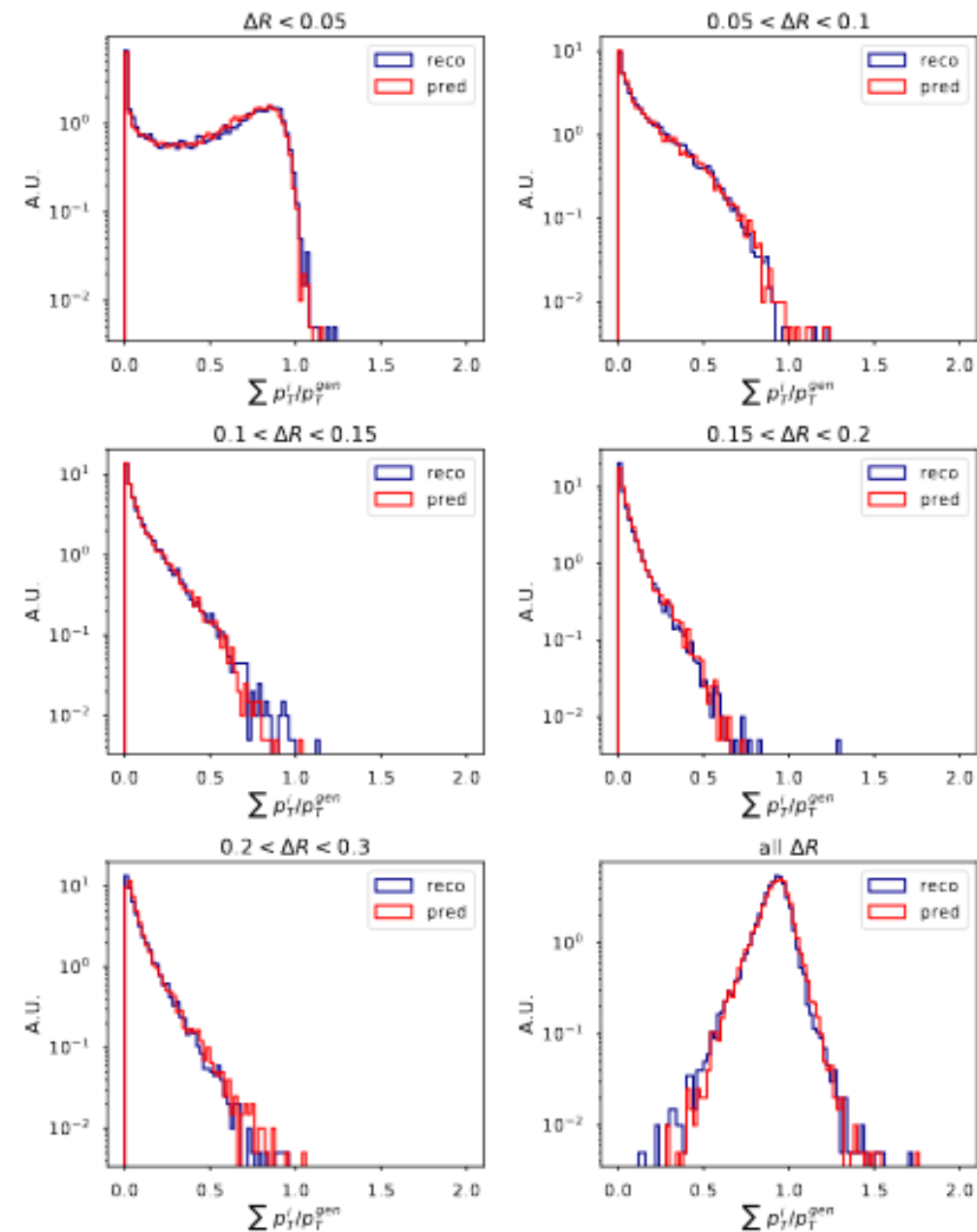
# GANs-Anwendung auf das Problem | Training

- Prozess analog zu Beispiel 1
- Als Input des Generators werden die Messdaten genutzt
- Als Datensatz werden die konventionell berechneten Bahnen verwendet
- Für das GANs Training wurde ein Supercomputer genutzt

Genutzt wurde der Daint Cray Supercomputer vom Swiss Centre for Supercomputing. Die GANs wurden mit 100Hz auf Intel Xeon CPUs und ca. 10kHz auf NVIDIA Pascal P100 GPUs berechnet. GANs wurden 10-20 Epochen lang trainiert. Das reichte aus, um Konvergenz zu erreichen. Eine Epochen betrug etwa eine Stunde.

# Resultate

- Blaue Kurve ist der Sollwert (konventionell berechnet)
- Rote Kurve sind die vom Generator erschaffenen Werte
- Im Durchschnitt sehr gute Ergebnisse



# Welchen Vorteil bringt es?

- Der Generator braucht nur einen Bruchteil der Zeit der konventionellen Methode
- Mehr Daten können verarbeitet werden
- Neue Methode, alte Ergebnisse
- Aber Approximation statt Berechnung

# GANs in der Wissenschaft - Bewertung

## Pro

- Leichte Nutzung
- Wenig Programmierung
- Unbeobachtetes Lernen
- dadurch relativ geringer (menschlicher) Zeitaufwand
- Methodik entwickelt sich schnell

## Contra

- Die meisten GANs-Anwendungen sind nicht wirklich sinnvoll in den Naturwissenschaften
- Alle Ausgaben/generierten Daten müssen überprüft werden (Realitätscheck)
- Suggestiert eventuell Korrektheit die nicht gegeben ist

# Zusammenfassung

- GANs sind eine neue Methodik
- Erfordern nicht viel Programmierung, aber Rechenzeit (in der Trainingsphase)
- Viele Anwendungsbereiche, manche nicht wirklich sinnvoll
- In Zukunft mit Sicherheit (noch) besser
- Anwendung in der Wissenschaft begrenzt, aber vorhanden

# Quellen 1

Allgemein zu GANs:

- <https://www.youtube.com/watch?v=9JpdAg6uMXs>
- <https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/>

Abbildungen:

- <https://becominghuman.ai/the-magic-of-generative-adversarial-network-gans-1c9d5da4a105>
- <https://mc.ai/generative-adversarial-network/>
- <https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/>



# Quellen 2 Papers

- <https://arxiv.org/pdf/1511.06434.pdf>
- <ps://arxiv.org/ftp/arxiv/papers/1802/1802.07228.pdf>
- <https://academic.oup.com/mnrasl/article/467/1/L110/2931732>
- <https://arxiv.org/abs/1805.00850v2>
- <https://arxiv.org/abs/1702.00403>
- <https://arxiv.org/pdf/1701.00160.pdf>
- <https://arxiv.org/abs/1710.10196v3>
- <https://arxiv.org/abs/1802.07228>
- <https://arxiv.org/abs/1406.2661>