* Ziel
  + Problem diffusion models: Simulieren einer markov chain braucht viel Zeit
  + Sie nutzen einen iterativen non-Markovian diffusion prozess, welcher viel schneller geht
* Consistency Property
  + DDIM haben neu ein consistency property so dass wenn man mit der gleichen latent variable startet, dass das daraus entstehende Bild ähnliche high-level features hat
  + Dies ermöglicht Interpolation im latent space (nicht nur pixel interpolation sondern semantic interpolation)
* Ich kann irgendwie direkter Zwischenschritte schätzen ohne alle Schritte durchgehen zu müssen
* Training bleibt gleich, nur sampling wird verändert. D.h. ich kann bestehende Models nutzen.
* Non-Markovian
  + Es gibt irgendwie andere forward processes, welche nicht mehr markovian sind
  + Bild hängt nicht nur von x\_t-1 ab sondern auch von x\_0
  + Ein Bild, das Text, Schrift, Reihe, Screenshot enthält.

    Automatisch generierte Beschreibung
  + Mit gewissen Design choices bleibt Beziehung zu originalem Bild x\_0 gleich
  + 
  + Jedoch Zwischenschritt zwischen t und t-1 ist neu
    - Ein Bild, das Schrift, Text, Reihe, Diagramm enthält.

      Automatisch generierte Beschreibung
    - Ein Bild, das Text, Schrift, Reihe, Handschrift enthält.

      Automatisch generierte Beschreibung
* Generation Process
  + Alte Formel DDPM
  + Ein Bild, das Schrift, Handschrift, Text, weiß enthält.

    Automatisch generierte Beschreibung
  + Neue Formel
  + Ein Bild, das Text, Schrift, Reihe, weiß enthält.

    Automatisch generierte Beschreibung
  + -> Gleiches model epsilon
* Eta
  + Eta skaliert sigma
  + Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

    Automatisch generierte Beschreibung
* Anzahl Schritte
  + Auch hier gilt, dass mehr Schritte bessere Qualität geben
  + Jedoch bereits ab 50 Schritte gute Qualität
  + Bei 1000 Schritte ist DDPM ein wenig besser als DDIM