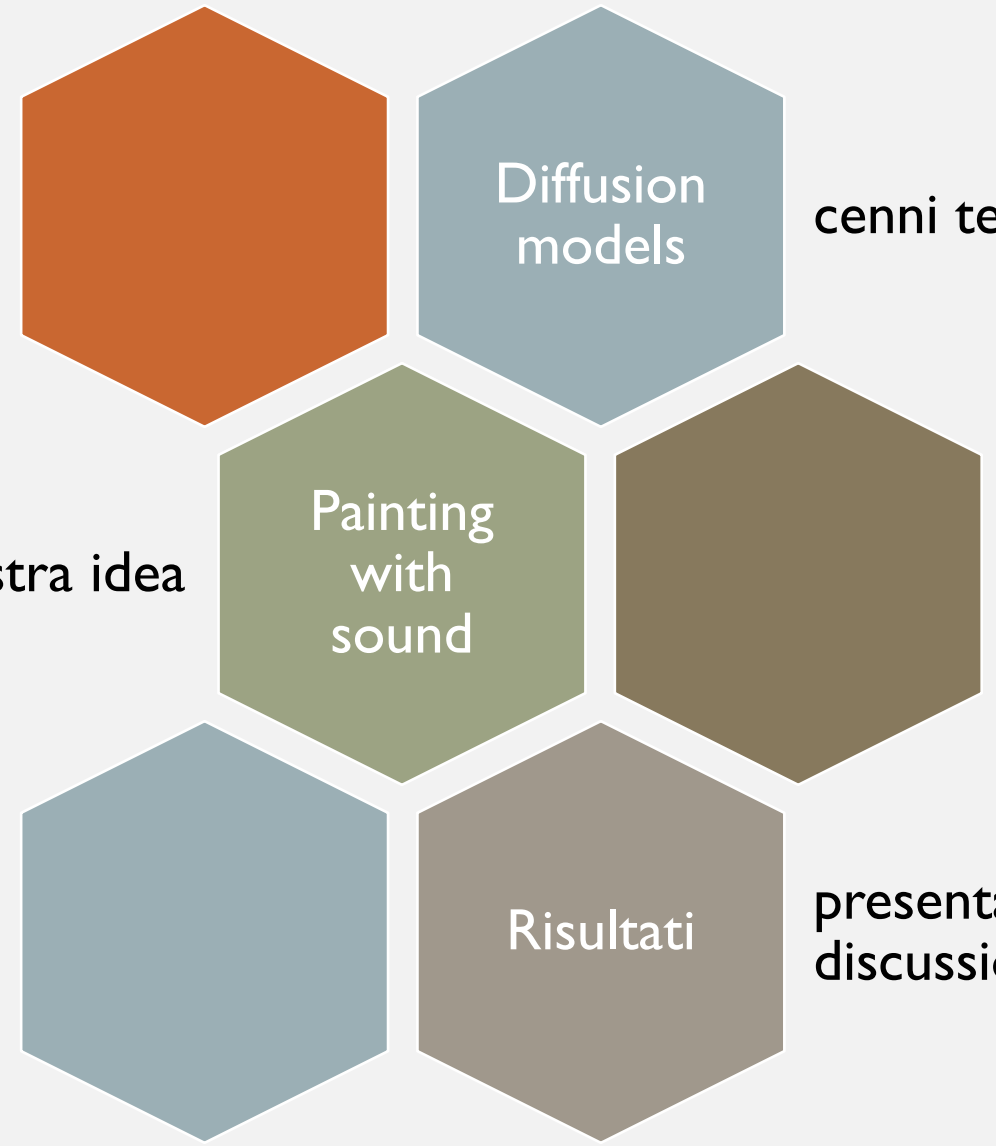
PAINTING WITH SOUND



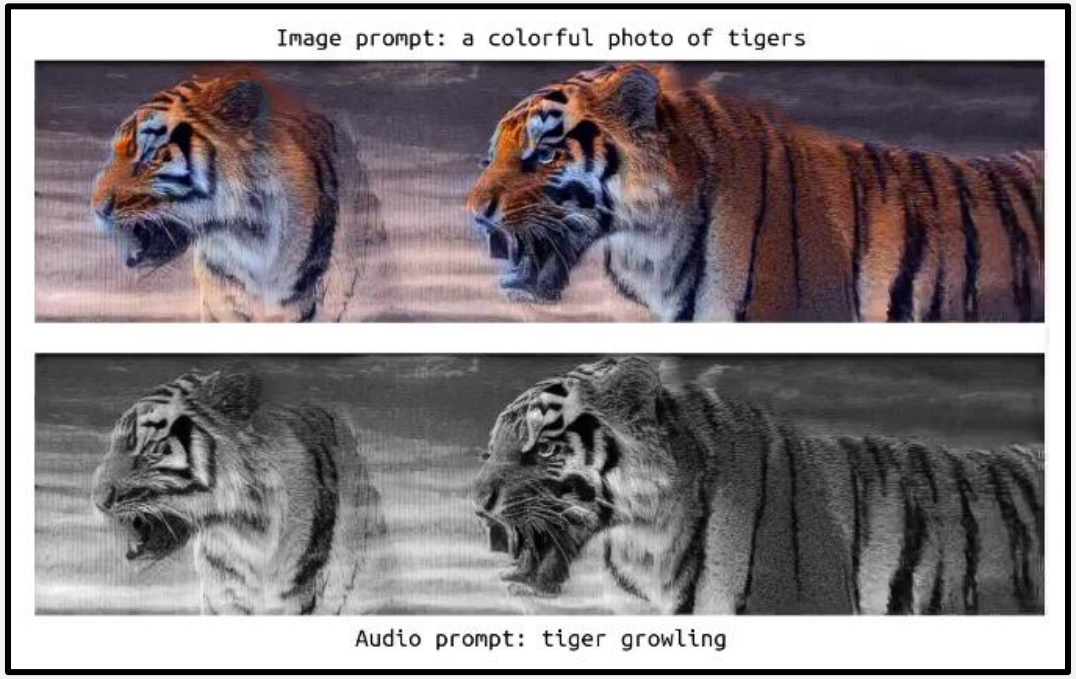
cenni teorici

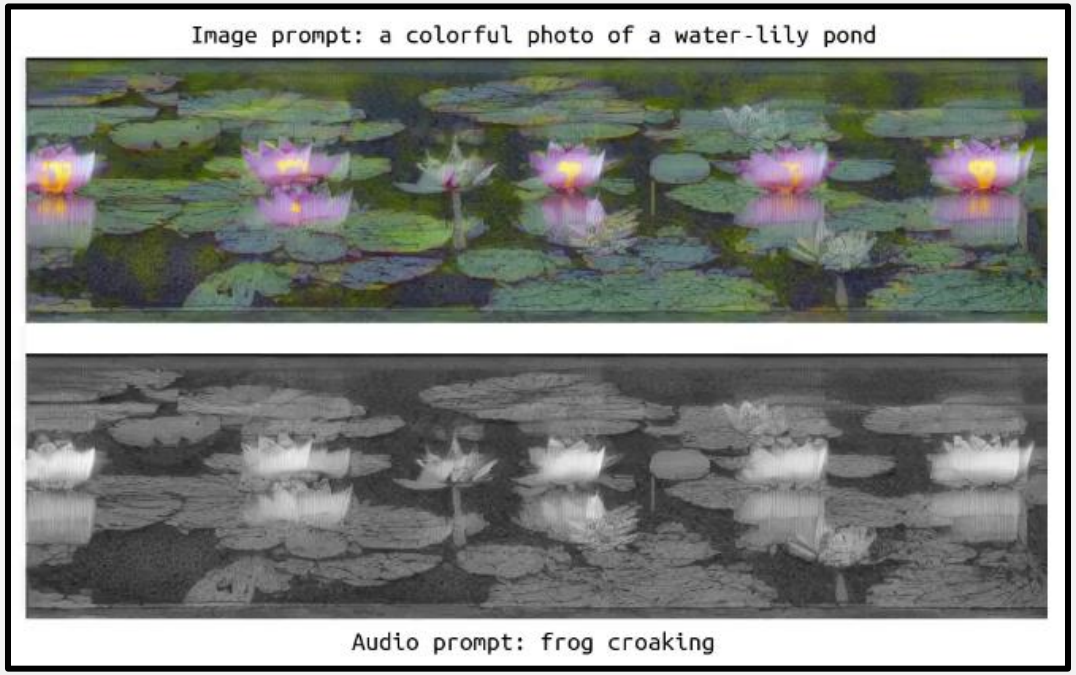
la nostra idea

presentazione e discussione

IMAGES THAT SOUND

«Gli spettrogrammi sono rappresentazioni bidimensionali del suono, ed appaiono molto diverse rispetto alle immagini presenti nel nostro mondo visivo. Inoltre, le immagini naturali, quando vengono riprodotte come spettrogrammi, producono suoni innaturali. In questo articolo, mostriamo che è possibile sintetizzare spettrogrammi che sembrano contemporaneamente immagini naturali e suonano come audio naturale. Chiamiamo questi spettrogrammi visivi immagini che suonano […]»





STABLE DIFFUSION

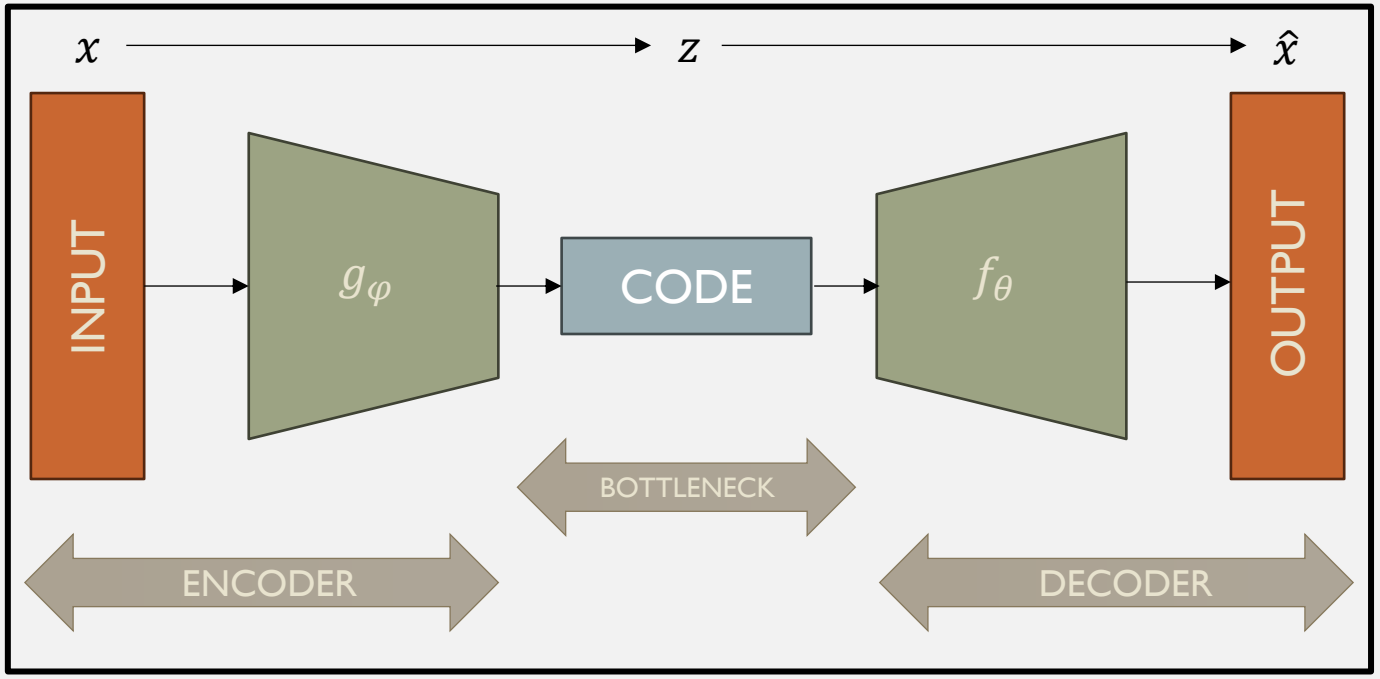
Stable diffusion è un modello deep learning text to image introdotto nel 2022 basato sui *diffusion models*.

Un modello generativo impara una distribuzione di probabilità del dataset, in modo da poter campionare da questa per generare nuovi dati.

COMPONENTI DI UN DIFFUSION MODEL

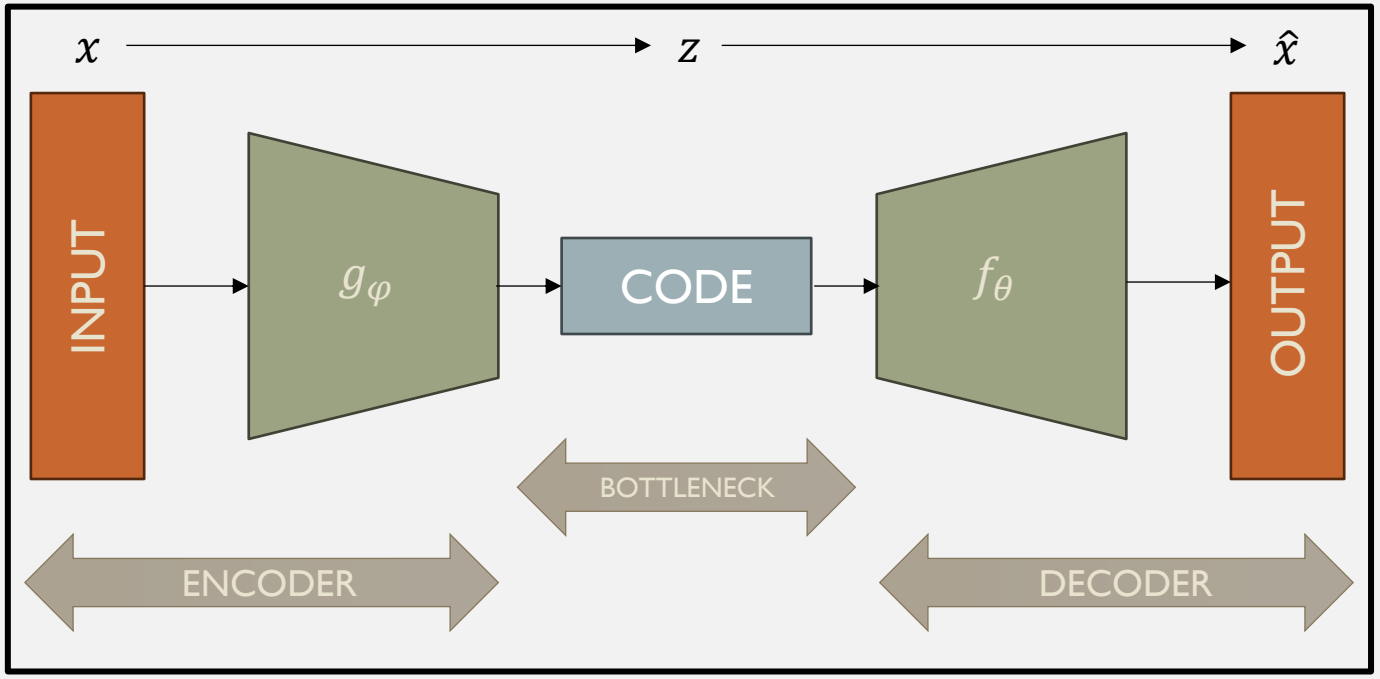
- VAE
- UNET
- TEXT ENCODER
- SCHEDULER

AUTOENCODERS



- Idealmente, $x \approx \hat{x}$;
- Il problema dell'autoencoder relativamente al problema della generazione di nuovi dati è che non impara le relazioni semantiche tra le codifiche;
- Nei modelli a diffusione sono utilizzati gli **autoencoder variazionali**.

VARIATIONAL AUTOENCODERS



- Nei VAE, l'encoder non restituisce un punto nello spazio latente, ma una distribuzione gaussiana normale nello spazio latente: $z \sim N(\mu, \sigma^2)$.
- Per addestrare il modello, si cerca di massimizzare la funzione **ELBO**:
  - Massimizzare la capacità di ricostruzione del decoder a partire da un campione;
  - Minimizzare la distanza che sussiste tra la z e una variabile aleatoria gaussiana multivariata.

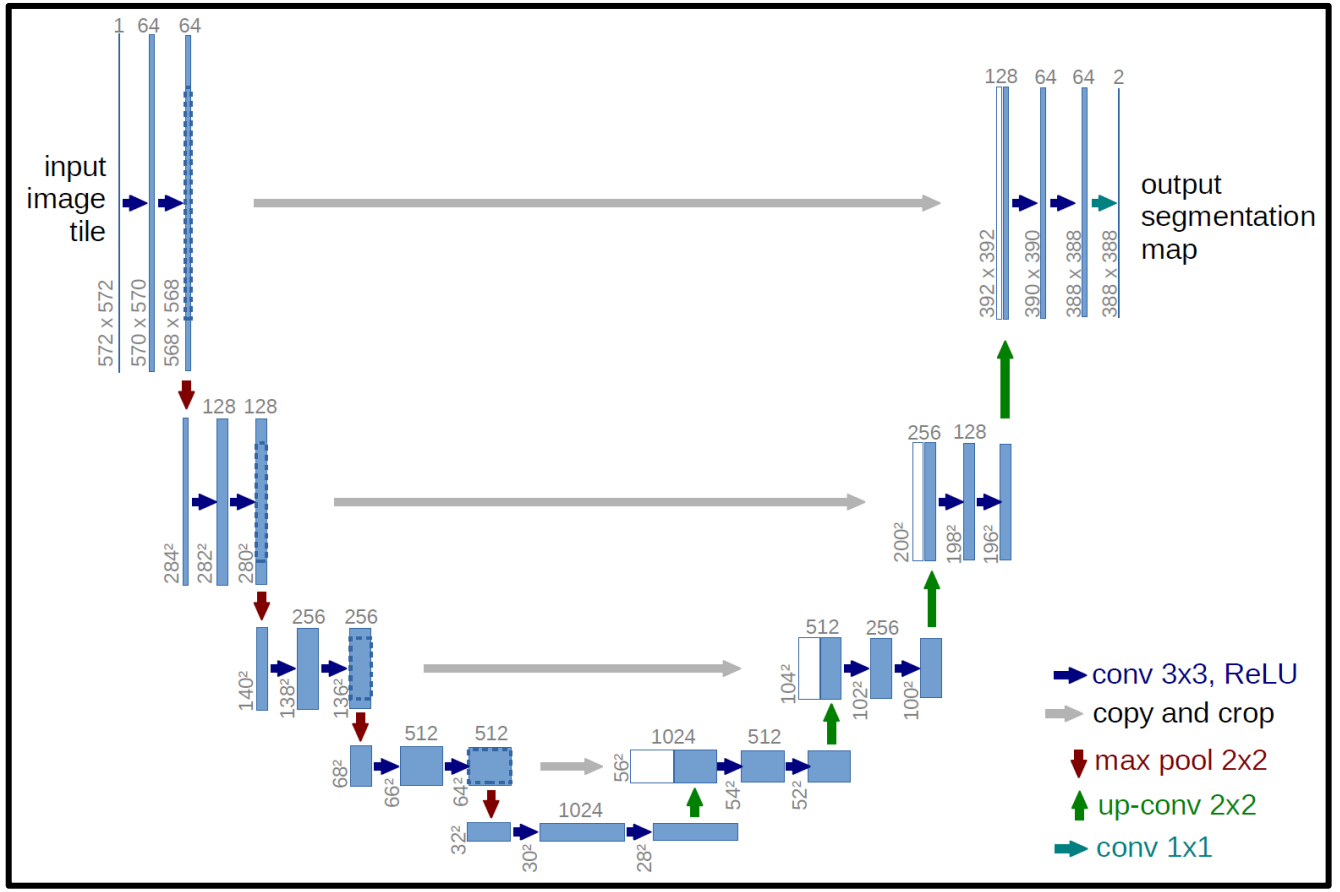
Lo spazio latente è più piccolo ma semanticamente più ricco. Per un modello di diffusione è semplice campionare dallo spazio latente e utilizzare il decoder per produrre nuovi dati.

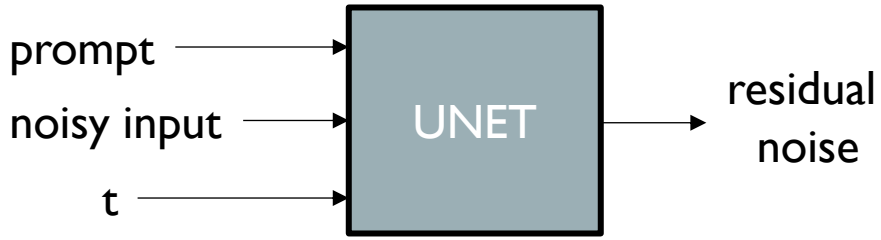
U-NET

Rete neurale convoluzionale di architettura encoder – decoder simmetrica con skip connections.

In questo contesto la U-NET è utilizzata sequenzialmente per predire ad ogni iterazione di denoising il rumore residuo.

Grazie alle skip-connections, il decoder può usare sia le feature **astratte** del bottleneck che le feature **locali** dell'encoder.





SCHEDULER

Componente la cui funzione è quella di processare il vettore latente, sia aggiungendo che rimuovendo iterativamente rumore

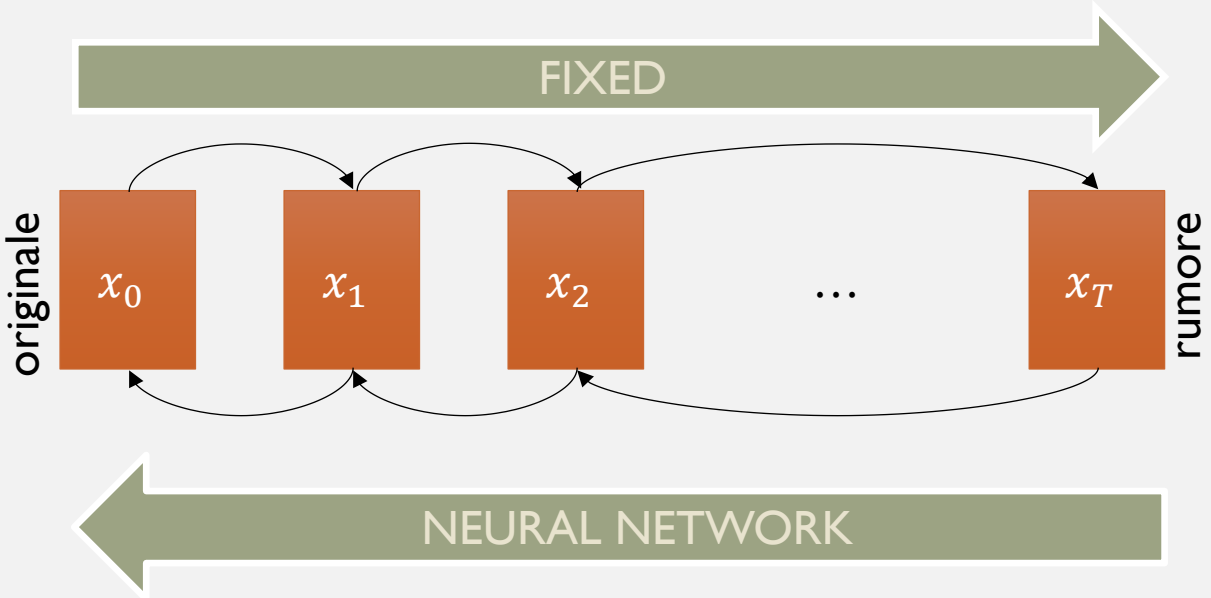
$$q(\boldsymbol{x_t}|\boldsymbol{x_{t-1}}) = N(\boldsymbol{x_t};\ \sqrt{1-\beta_t}\boldsymbol{x_{t-1}}, \beta_t \boldsymbol{I})$$

$\beta_1, \beta_2, \dots, \beta_T$ coefficienti definiti dal progettista

$$q(\boldsymbol{x_t}|\boldsymbol{x_0}) = N(\boldsymbol{x_t};\ \sqrt{\overline{\alpha_t}}\boldsymbol{x_0}, (1-\overline{\alpha_t})\boldsymbol{I})$$

$$\alpha_t = 1 - \beta_t$$

$$\overline{\alpha_t} = \prod_{s=1}^{t} \alpha_s$$

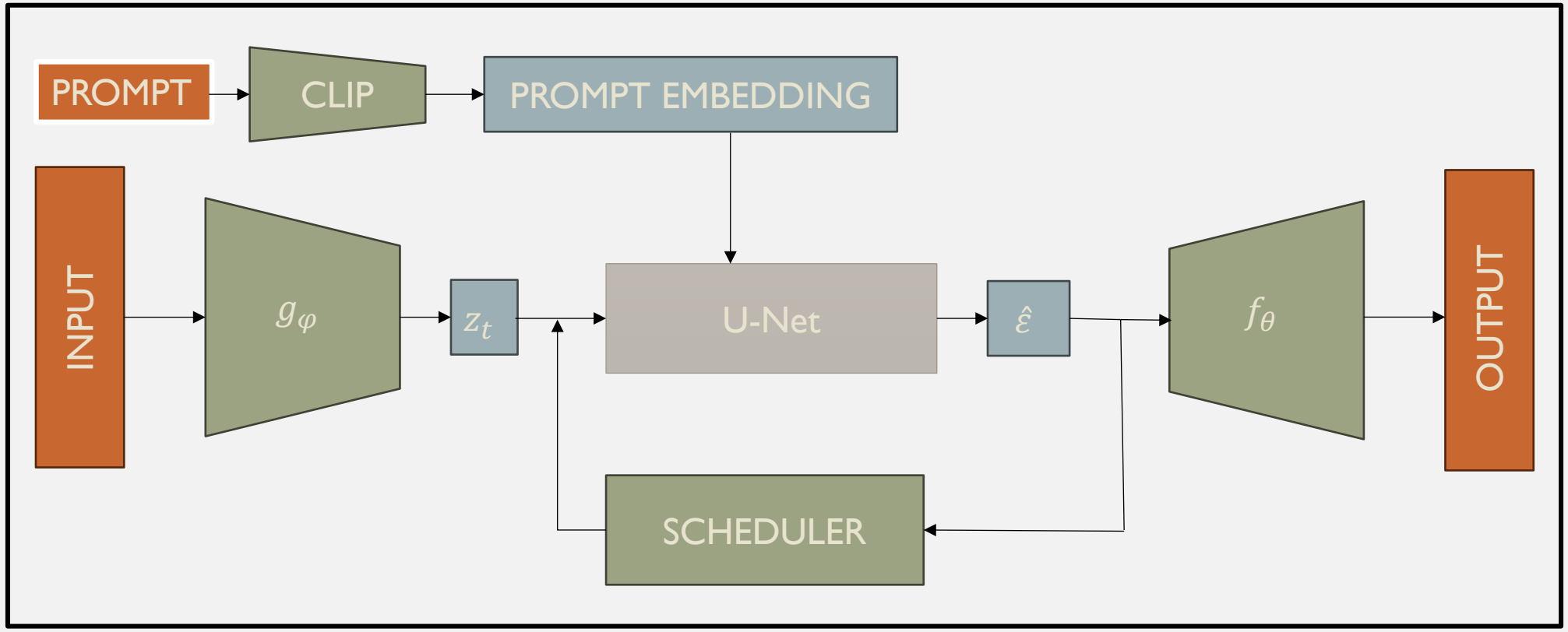


$$\boldsymbol{x_t} = \sqrt{\overline{\alpha_t}}\boldsymbol{x_0} + \sqrt{(1-\overline{\alpha_t})}\ \underline{\boldsymbol{I}} * \underline{\boldsymbol{noise}}$$

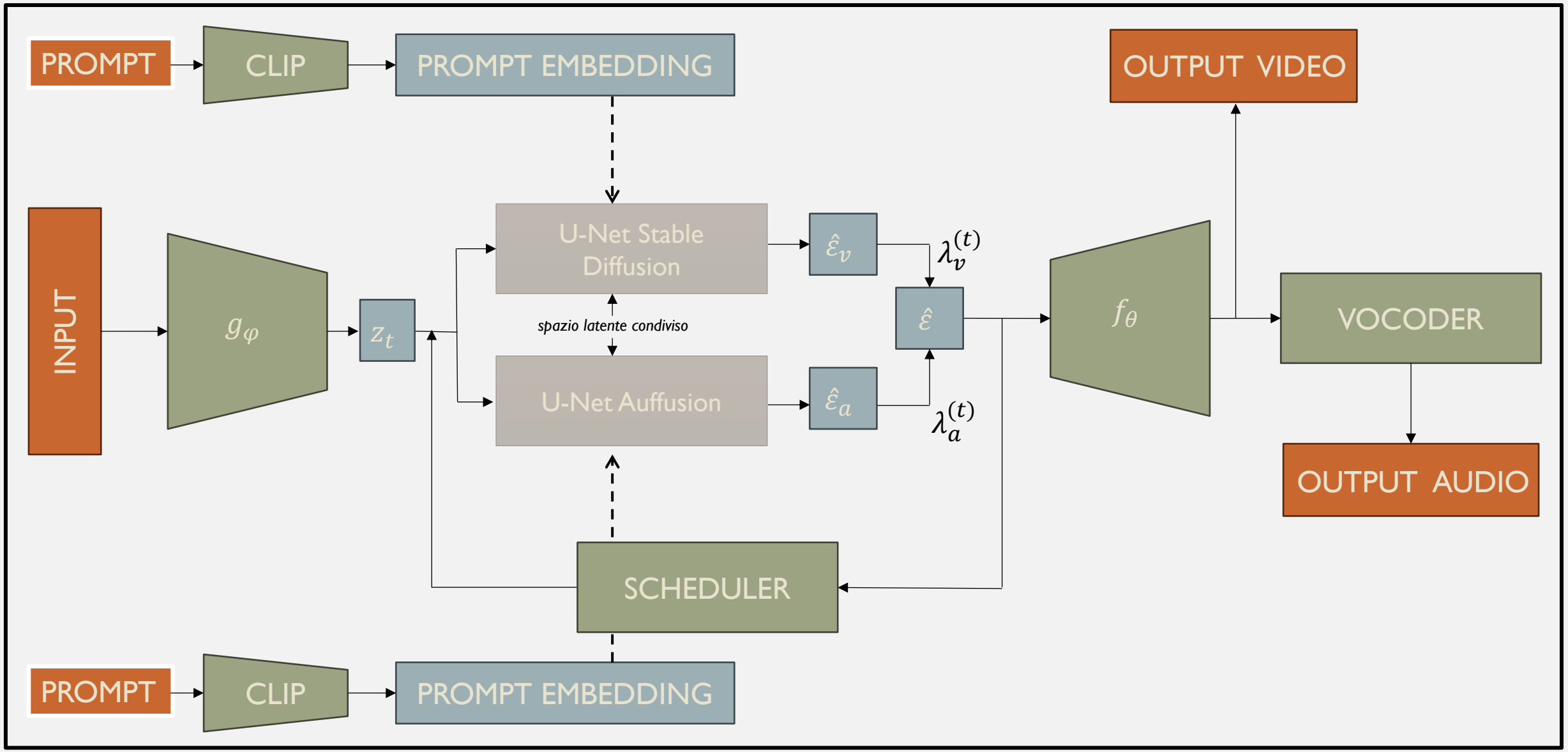
$$\boldsymbol{x_{t-1}} = \frac{1}{\sqrt{\overline{\alpha_t}}}\left(\boldsymbol{x_t} - \frac{\beta_t}{\sqrt{1-\overline{\alpha_t}}}\boldsymbol{\varepsilon_\theta}(x_t, t)\right)$$

$\boldsymbol{\varepsilon_\theta}(x_t, t)$ è il rumore stimato dalla UNET

ARCHITETTURA STABLE DIFFUSION



ARCHITETTURA DI RIFERIMENTO

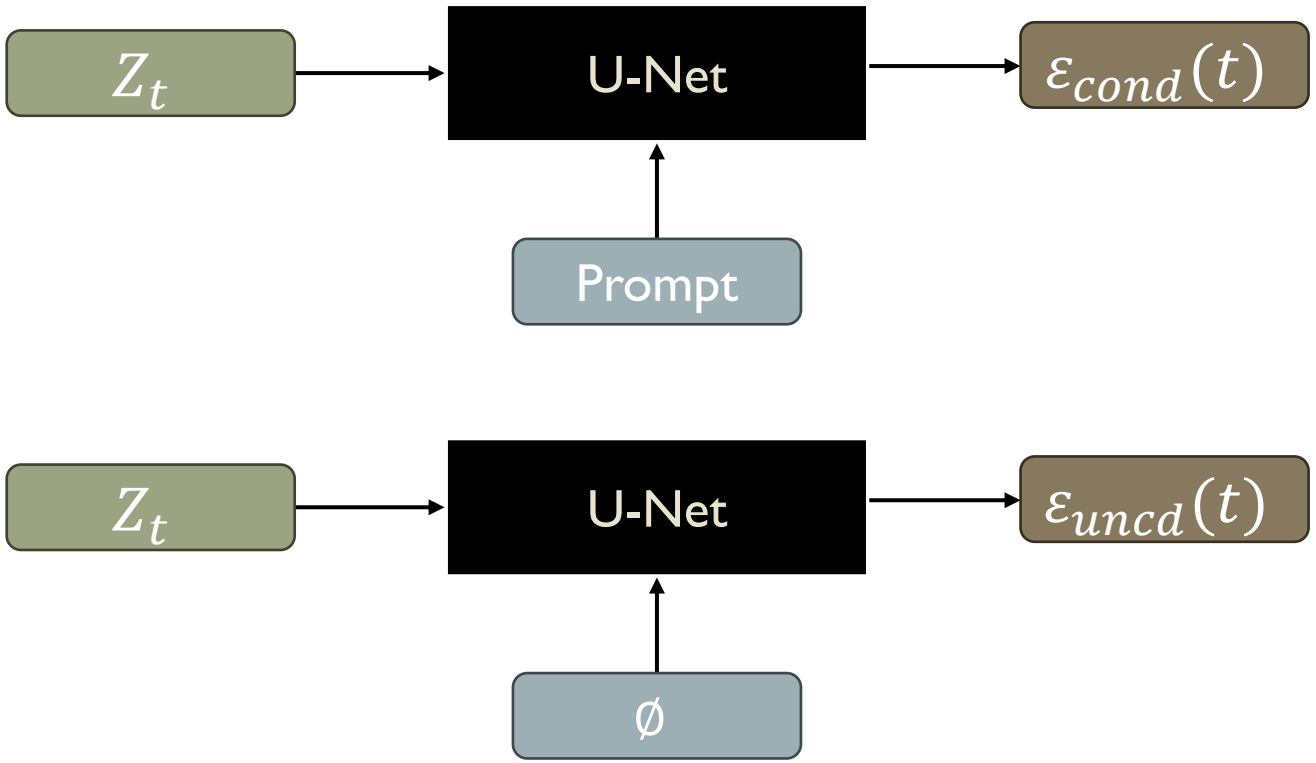


GUIDANCE SCALE

I prompt incidono in fase di denoising, anche se si può impostare **quanto incidano**

Per ogni passo di denoising, calcoliamo sia un rumore **condizionato** che uno **non condizionato** (dal prompt)

Una volta «calcolati» si combinano utilizzando il parametro **guidance scale (γ)**



$$\varepsilon_x(t) = \varepsilon_{uncd}(t) + \gamma\left(\varepsilon_{cond}(t) - \varepsilon_{uncd}(t)\right)$$

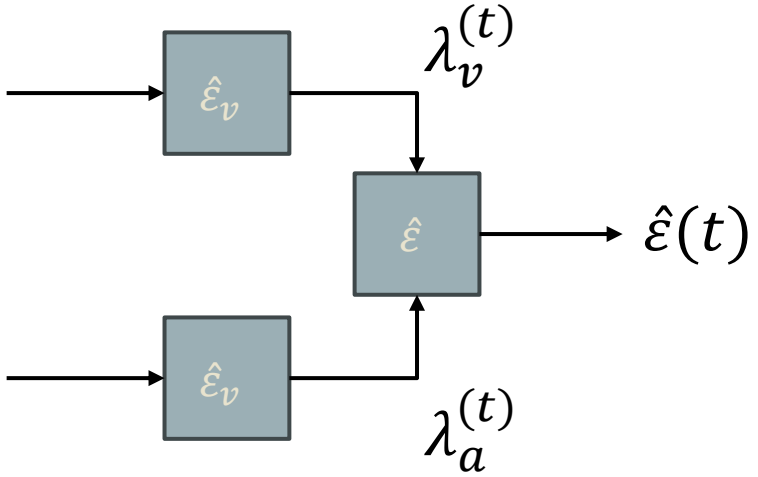
$$x \in \{a, v\} \rightarrow tipologia\ di\ U-Net$$

AGGREGAZIONE DEI RUMORI

Date le uscite prodotte dalle UNET, queste dovranno essere opportunamente combinate

I lambda dipendono da due coefficienti: $\boldsymbol{w_a}$ e $\boldsymbol{w_v}$

In generale tale prodotto è una **media pesata** che, in quasi tutto il suo ciclo di vita non è altro che una media aritmetica



$$\lambda_v(t) = \frac{w_v(t)}{w_v(t) + w_a(t)}$$

$$\lambda_a(t) = \frac{w_a(t)}{w_v(t) + w_a(t)}$$

$$\hat{\varepsilon}(t) = \lambda_a(t) * \hat{\varepsilon}_a(t) + \lambda_v(t) * \hat{\varepsilon}_v(t)$$

AGGREGAZIONE DEI RUMORI

La media pesata permette di considerare quale componente rifinire maggiormente

Tale sistema permette di **«interrompere»** la definizione di alcuni **dettagli** nei primi cicli di denoising

I parametri potrebbero essere anche appresi, ma richiederebbero un **dataset** ad-hoc e di risorse **hardware maggiori**

$$\lambda_v(t) = \frac{w_v(t)}{w_v(t) + w_a(t)}$$

$$\lambda_a(t) = \frac{w_a(t)}{w_v(t) + w_a(t)}$$

$$\hat{\varepsilon}(t) = \lambda_a(t) * \hat{\varepsilon}_a(t) + \lambda_v(t) * \hat{\varepsilon}_v(t)$$

$$w_v(t) = H(t_v * T - t)$$

$$w_a(t) = H(t_a * T - t)$$

$$t_a, t_v \in [0,1]$$

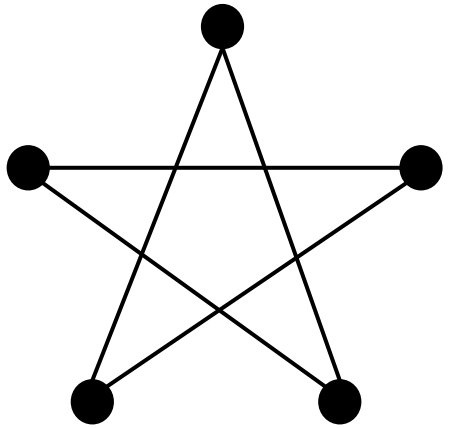
$$H(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases}$$

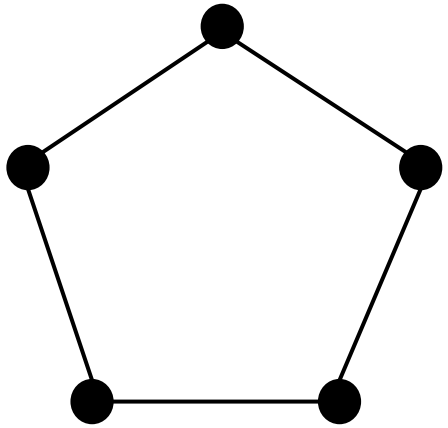
GENERAZIONE GUIDATA

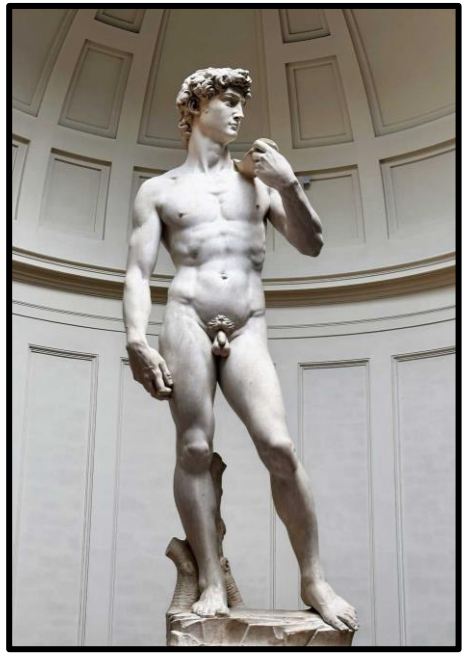
Il sistema precedente richiedeva la generazione di **50 immagini** e di selezionare il miglior risultato

Ogni immagine impiega circa una **40 secondi**…

Magari non partire da **«un foglio bianco»,** ma da un sistema **«puntineggiante»**





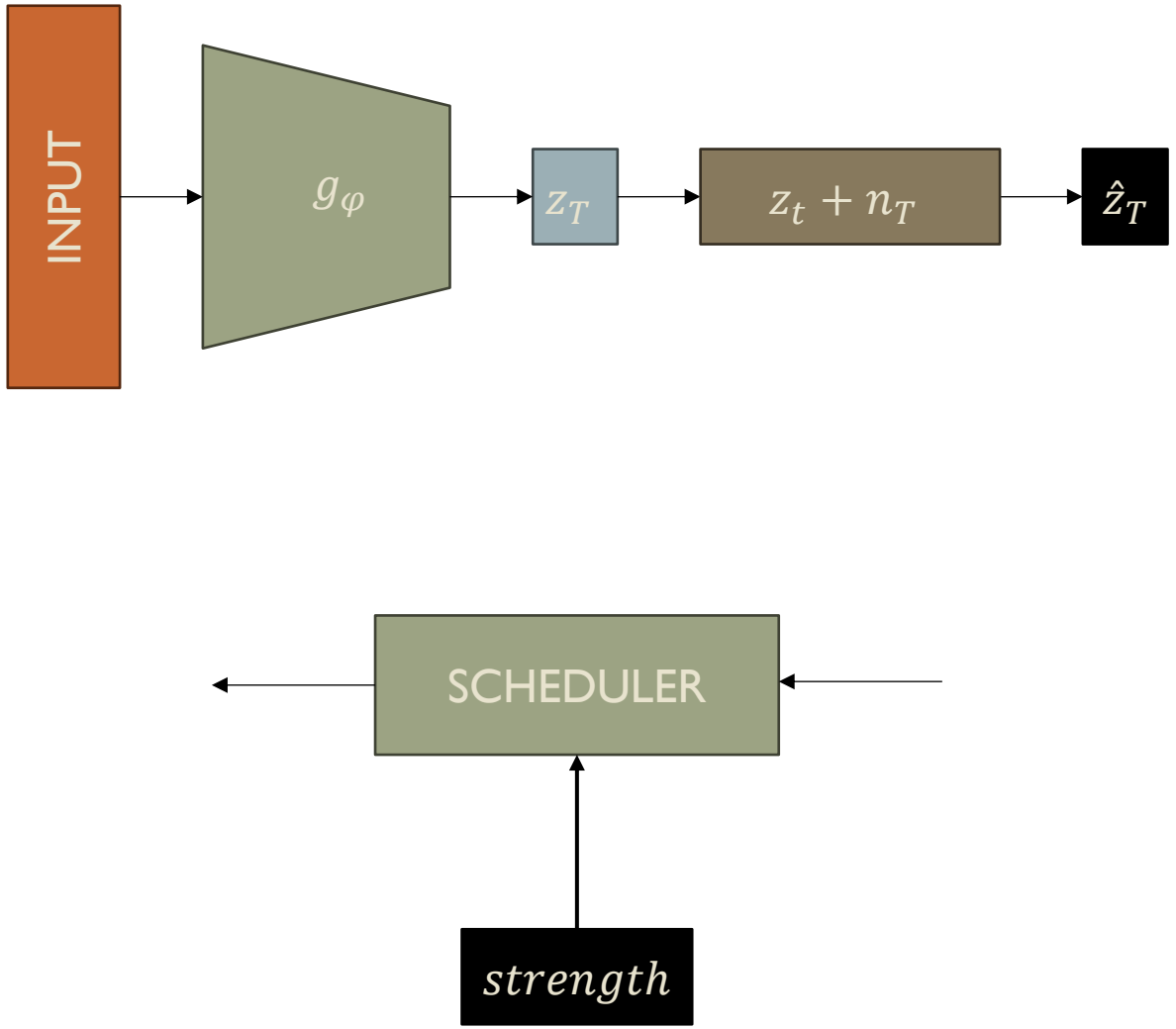


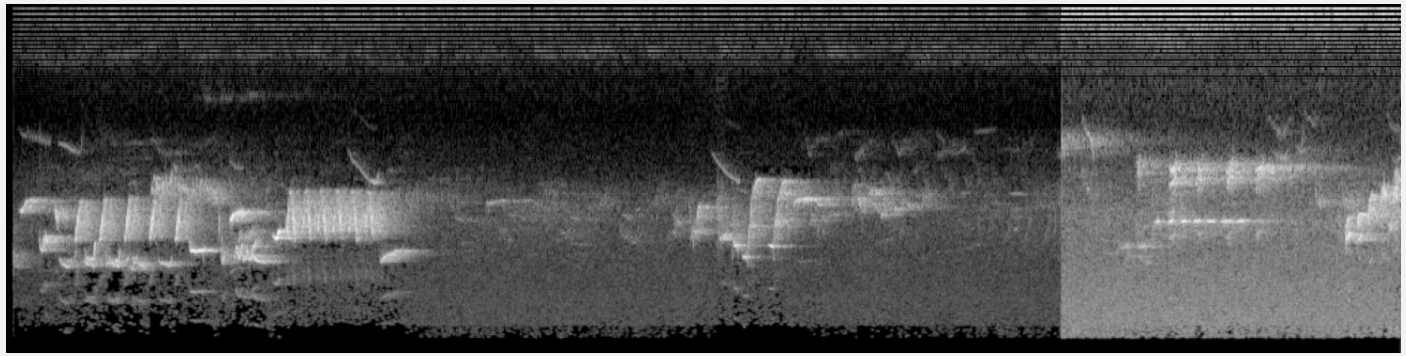
GENERAZIONE GUIDATA

L' «impronta» viene sporcata e poi inserita come **vettore iniziale** in fase di denoising

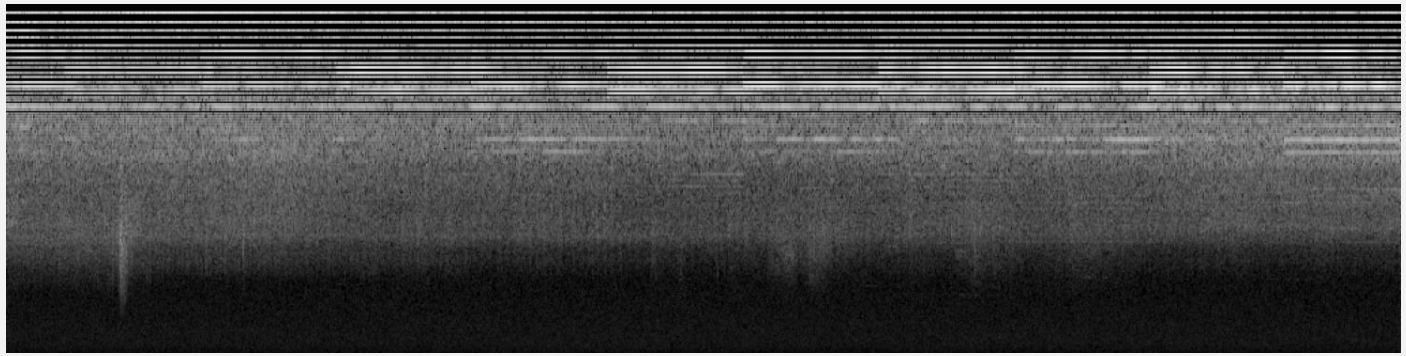
Oltre al vettore iniziale bisogna anche decidere di quanto tale vettore **incida**

Il parametro permette di saltare vari passi di denoising iniziali **enfatizzando** lo spettrogramma di partenza

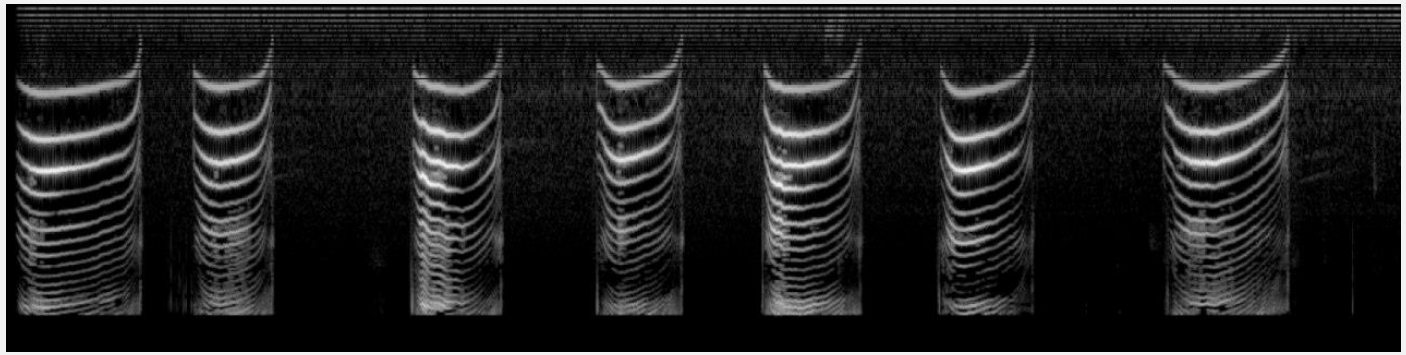




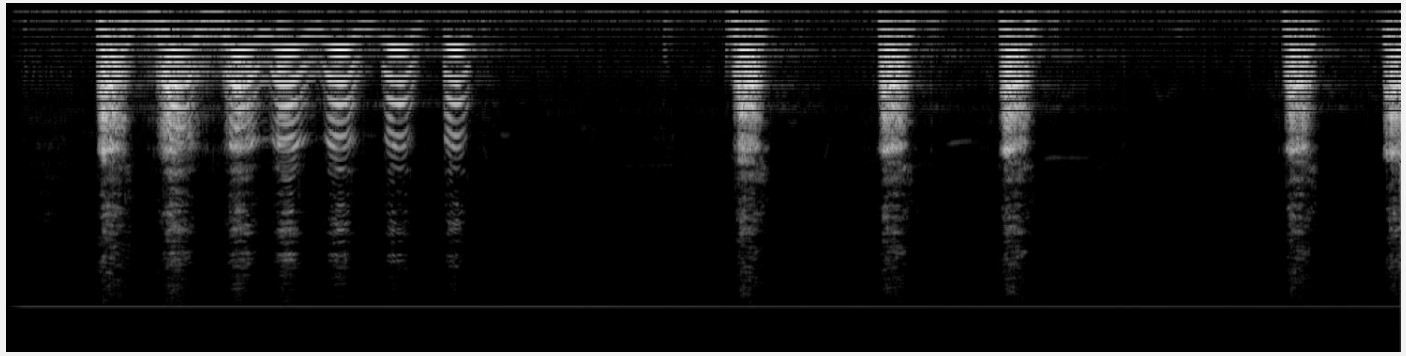
Prompt video	sea landscape with ships, grayscale
Prompt audio	birds singing softly
Guidance scale audio	10.0
Guidance scale video	8.0
Strength	0.8



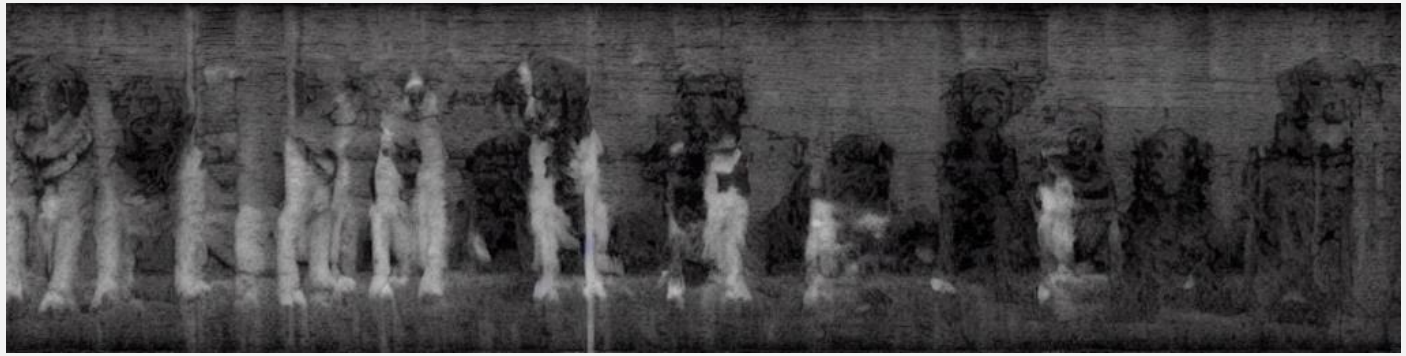
Prompt video	a painting of police cars, grayscale
Prompt audio	police siren
Guidance scale audio	7.0
Guidance scale video	9.5
Strength	0.8



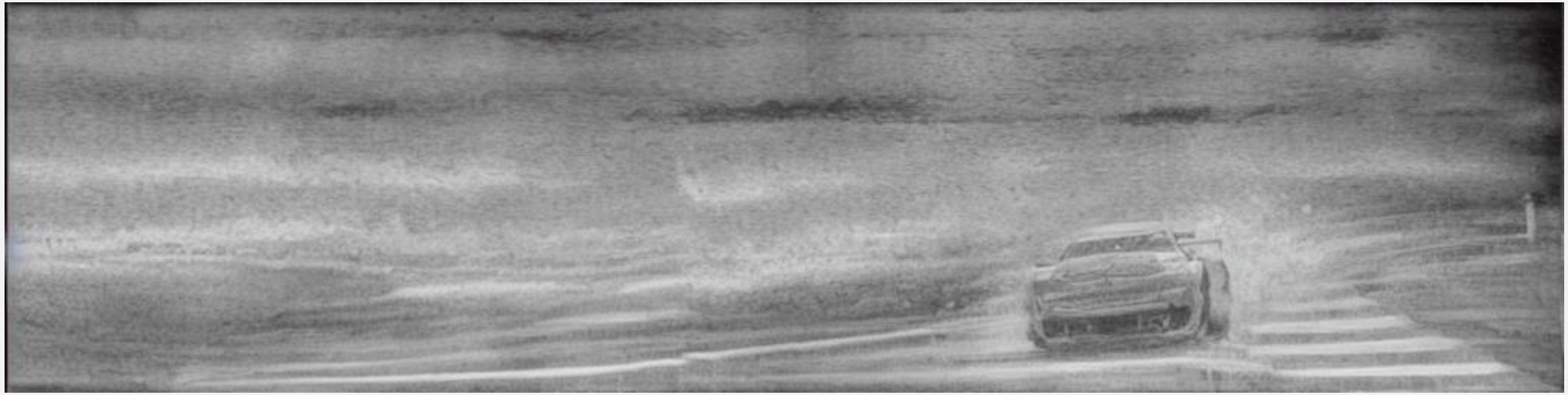
Prompt video	a series of column of a Greece temple, grayscale
Prompt audio	kitten meowing for attention
Guidance scale audio	7.0
Guidance scale video	9.5
Strength	0.8



Prompt video	columns of a Greece temple, grayscale
Prompt audio	a dog barking
Guidance scale audio	6.0
Guidance scale video	9.5
Strength	0.9



Prompt video	dogs, grayscale
Prompt audio	dogs barking, clean audio
Guidance scale audio	10.0
Guidance scale video	7.0
Strength	0.8



Prompt video	A picture of sport racing car, grayscale
Prompt audio	Car drifting
Guidance scale audio	7.5
Guidance scale video	8

PAINTING WITH SOUND 1.2

Implementare il supporto agli spettrogrammi di voci umane

Auffusion ++

Fine tuning

Vocoder ++

Training di iperparametri interni

Painting with music

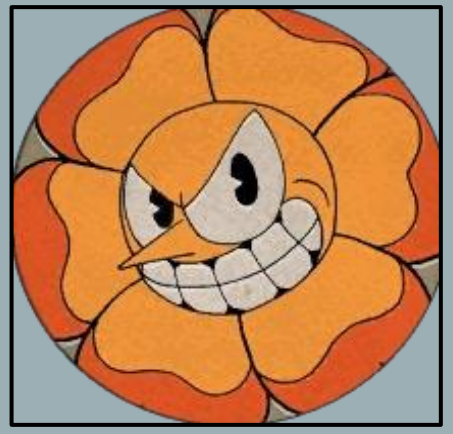
BIBLIOGRAFIA

- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10684-10695).

- Xue, J., Deng, Y., Gao, Y., & Li, Y. (2024). Auffusion: Leveraging the power of diffusion and large language models for text-to-audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

- Chen, Z., Geng, D., & Owens, A. (2024). Images that sound: Composing images and sounds on a single canvas. *Advances in Neural Information Processing Systems*, *37*, 85045-85073.

TEAM H-TAJATO



Rocco Lo Russo
Studente

Agostino D'Amora
Studente

GRAZIE PER L'ATTENZIONE