Unsupervised Speech Enhancement with Diffusion-based Generative Models

Berné Nortier, Mostafa Sadeghi & Romain Serizel

Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France

2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2024)

April 14-19, Seoul, Korea.

What is speech enhancement?

- In practice, speech is recorded in noisy environments $\rightarrow$ **speech enhancement** (SE)

SE: Given noisy speech observation $\mathbf{x} = \mathbf{s} + \mathbf{n}$, estimate the clean speech signal $\mathbf{s}$.

What is speech enhancement?

- In practice, speech is recorded in noisy environments → **speech enhancement** (SE)

SE: Given noisy speech observation $\mathbf{x} = \mathbf{s} + \mathbf{n}$, estimate the clean speech signal $\mathbf{s}$.

- Complex-valued short-time Fourier transform domain

Approaches to SE

- **Supervised:** Model $p_{\Theta}(\mathbf{s}|\mathbf{x})$

Approaches to SE

▷ **Supervised:** Model $p_\Theta(\mathbf{s}|\mathbf{x})$, and learn Θ

- Train on pairs of noisy-clean data $\{\mathbf{x}_i, \mathbf{s}_i\}$

Approaches to SE

▷ **Supervised:** Model $p_\Theta(\mathbf{s}|\mathbf{x})$, and learn Θ

- Train on pairs of noisy-clean data $\{\mathbf{x}_i, \mathbf{s}_i\}$
- *Implicit* prior modelling $p_\theta(\mathbf{s})$ via inductive biases (architecture, optimizer, etc.)

☞ Cannot cover all test cases

Approaches to SE

▷ **Supervised:** Model $p_\Theta(\mathbf{s}|\mathbf{x})$, and learn Θ

- Train on pairs of noisy-clean data $\{\mathbf{x}_i, \mathbf{s}_i\}$
- *Implicit* prior modelling $p_\theta(\mathbf{s})$ via inductive biases (architecture, optimizer, etc.)

☞ Cannot cover all test cases

▷ **Unsupervised:**

- Training - - *only* on clean speech signals

Approaches to SE

▷ **Supervised:** Model $p_\Theta(\mathbf{s}|\mathbf{x})$, and learn Θ

- Train on pairs of noisy-clean data $\{\mathbf{x}_i, \mathbf{s}_i\}$
- *Implicit* prior modelling $p_\theta(\mathbf{s})$ via inductive biases (architecture, optimizer, etc.)

☞ Cannot cover all test cases

▷ **Unsupervised:**

- Training - Learn speech's prior distribution $p_\theta(\mathbf{s})$- *only* on clean speech signals

Approaches to SE

▷ **Supervised:** Model $p_\Theta(\mathbf{s}|\mathbf{x})$, and learn Θ

- Train on pairs of noisy-clean data $\{\mathbf{x}_i, \mathbf{s}_i\}$
- *Implicit* prior modelling $p_\theta(\mathbf{s})$ via inductive biases (architecture, optimizer, etc.)

☞ Cannot cover all test cases

▷ **Unsupervised:** Model $p_\Theta(\mathbf{s}|\mathbf{x}) \propto \underbrace{p_\phi(\mathbf{x}|\mathbf{s})}_{\text{Inference}} \cdot \underbrace{p_\theta(\mathbf{s})}_{\text{Training}}$, and learn $\Theta = \theta \cup \phi$

- Training - Learn speech's prior distribution $p_\theta(\mathbf{s})$- *only* on clean speech signals

Approaches to SE

- **Supervised:** Model $p_\Theta(\mathbf{s}|\mathbf{x})$, and learn Θ
  - Train on pairs of noisy-clean data $\{\mathbf{x}_i, \mathbf{s}_i\}$
  - *Implicit* prior modelling $p_\theta(\mathbf{s})$ via inductive biases (architecture, optimizer, etc.)

☞ Cannot cover all test cases

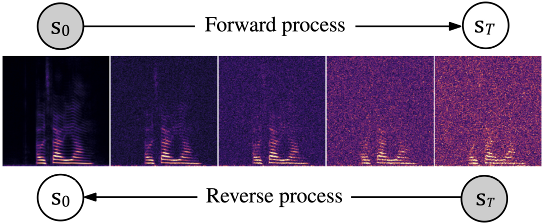
- **Unsupervised:** Model $p_\Theta(\mathbf{s}|\mathbf{x}) \propto \underbrace{p_\phi(\mathbf{x}|\mathbf{s})}_{\text{Inference}} \cdot \underbrace{p_\theta(\mathbf{s})}_{\text{Training}}$, and learn $\Theta = \theta \cup \phi$
  - Training - Learn speech's prior distribution $p_\theta(\mathbf{s})$- *only* on clean speech signals
  - Inference - Model $p_\phi(\mathbf{x}|\mathbf{s})$, and infer $\mathbf{s}$ using $p_\theta(\mathbf{s})$

☞ May offer superior generalization

Score-based generative models for SE

▷ **Previous (supervised) diffusion-based work: SMGSE+** [1]

- Gradually corrupt clean speech with both Gaussian and environmental noise

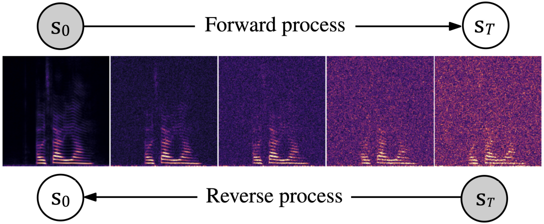


[1] J. Richter *et al.*, "Speech enhancement and dereverberation with diffusion-based generative models," IEEE/ACM TASLP, 2023.

Score-based generative models for SE

▷ **Previous (supervised) diffusion-based work: SMGSE+** [1]

- Gradually corrupt clean speech with both Gaussian and environmental noise
- Learn a neural network (*conditional* **score model**) to revert the process

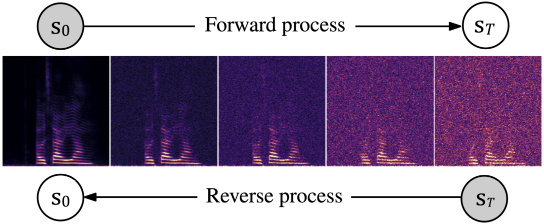


[1] J. Richter *et al.*, "Speech enhancement and dereverberation with diffusion-based generative models," IEEE/ACM TASLP, 2023.

Score-based generative models for SE

▷ **Previous (supervised) diffusion-based work: SMGSE+** [1]

- Gradually corrupt clean speech with both Gaussian and environmental noise
- Learn a neural network (*conditional* **score model**) to revert the process



- Processes can be modelled as a *Stochastic Differential Equation (SDE)*

[1] J. Richter *et al.*, "Speech enhancement and dereverberation with diffusion-based generative models," IEEE/ACM TASLP, 2023.

Unsupervised diffusion-based SE

Estimate clean speech **s** by directly sampling from the **intractable** posterior:

$$p_{\Theta}(\mathbf{s}|\mathbf{x}) \propto p_{\phi}(\mathbf{x}|\mathbf{s})p_{\theta}(\mathbf{s})$$

Unsupervised diffusion-based SE

Estimate clean speech $\mathbf{s}$ by directly sampling from the **intractable** posterior:

$$p_{\Theta}(\mathbf{s}|\mathbf{x}) \propto p_{\phi}(\mathbf{x}|\mathbf{s})p_{\theta}(\mathbf{s})$$

▷ **UDiffSE** modelling framework:

- $p_{\theta}(\mathbf{s})$: Learn via an *unconditional* diffusion model

Unsupervised diffusion-based SE

Estimate clean speech $\mathbf{s}$ by directly sampling from the **intractable** posterior:

$$p_\Theta(\mathbf{s}|\mathbf{x}) \propto p_\phi(\mathbf{x}|\mathbf{s})p_\theta(\mathbf{s})$$

▷ **UDiffSE** modelling framework:

- $p_\theta(\mathbf{s})$: Learn via an *unconditional* diffusion model
- $p_\phi(\mathbf{x}|\mathbf{s})$: Model noise $\mathbf{n} \sim \mathcal{N}_\mathbb{C}\left(\mathbf{0}, \mathrm{diag}(\mathbf{v}_\phi)\right)$

Unsupervised diffusion-based SE

Estimate clean speech $\mathbf{s}$ by directly sampling from the **intractable** posterior:

$$p_\Theta(\mathbf{s}|\mathbf{x}) \propto p_\phi(\mathbf{x}|\mathbf{s})p_\theta(\mathbf{s})$$

▷ **UDiffSE** modelling framework:

- $p_\theta(\mathbf{s})$: Learn via an *unconditional* diffusion model
- $p_\phi(\mathbf{x}|\mathbf{s})$: Model noise $\mathbf{n} \sim \mathcal{N}_\mathbb{C}\Big(\mathbf{0}, \mathrm{diag}(\mathbf{v}_\phi)\Big)$

$$p_\phi(\mathbf{x}|\mathbf{s}) = \mathcal{N}_\mathbb{C}\Big(\mathbf{s}, \mathrm{diag}(\mathbf{v}_\phi)\Big)$$

Unsupervised diffusion-based SE

Estimate clean speech $\mathbf{s}$ by directly sampling from the **intractable** posterior:

$$p_\Theta(\mathbf{s}|\mathbf{x}) \propto p_\phi(\mathbf{x}|\mathbf{s})p_\theta(\mathbf{s})$$

▷ **UDiffSE** modelling framework:

- $p_\theta(\mathbf{s})$: Learn via an *unconditional* diffusion model
- $p_\phi(\mathbf{x}|\mathbf{s})$: Model noise $\mathbf{n} \sim \mathcal{N}_\mathbb{C}\left(\mathbf{0}, \mathrm{diag}(\mathbf{v}_\phi)\right)$

$$p_\phi(\mathbf{x}|\mathbf{s}) = \mathcal{N}_\mathbb{C}\left(\mathbf{s}, \mathrm{diag}(\mathbf{v}_\phi)\right)$$

  - $\mathbf{v}_\phi = \mathrm{vec}(\mathbf{WH}) \leftarrow$ non-negative matrix factorisation (NMF)

Inference framework: Expectation-maximisation

▷ Iterative **Expectation Maximisation**-based inference ($k = 1, \ldots, K$):

1. **E-step:** *Draw posterior sample*

$$\hat{\mathbf{s}}_k \sim p_{\Theta_{k-1}}(\mathbf{x}|\mathbf{s}) \quad \rightarrow \text{reverse diffusion}$$

Inference framework: Expectation-maximisation

▷ Iterative **Expectation Maximisation**-based inference $(k = 1, \dots, K)$:

1. **E-step:** *Draw posterior sample*

$$\hat{\mathbf{s}}_k \sim p_{\Theta_{k-1}}(\mathbf{x}|\mathbf{s}) \quad \rightarrow \text{reverse diffusion}$$

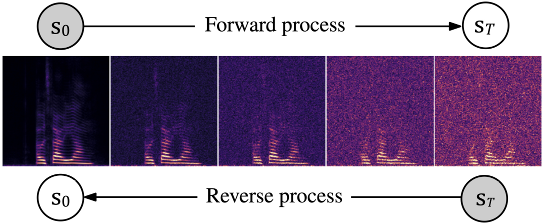
2. **M-step:** *Maximise likelihood*

$$\phi_k \leftarrow \underset{\phi}{\text{argmax}} \; \log p_\phi(\mathbf{x}|\hat{\mathbf{s}}_k) \quad \rightarrow \text{NMF update}$$

Prior: Diffusion-based speech generative model

▷ *Unconditional* (prior) diffusion model for complex-valued clean speech STFT:

- **Noising (forward) SDE:** [2] $\mathrm{d}\mathbf{s}_t = \mathbf{f}(\mathbf{s}_t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \quad \mathbf{f}(\mathbf{s}_t) = -\gamma \mathbf{s}_t$

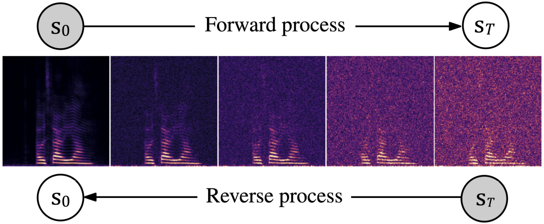


[2] Y. Song *et al.*, "Score-based generative modelling through stochastic differential equations", ICLR, 2021.

Prior: Diffusion-based speech generative model

▷ *Unconditional* (prior) diffusion model for complex-valued clean speech STFT:

- **Noising (forward) SDE:** [2] $\mathrm{d}\mathbf{s}_t = \mathbf{f}(\mathbf{s}_t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \quad \mathbf{f}(\mathbf{s}_t) = -\gamma \mathbf{s}_t$



- **Denoising (reverse) SDE:** $\mathrm{d}\mathbf{s}_t = [\mathbf{f}(\mathbf{s}_t) - g(t)^2 \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$

[2] Y. Song *et al.*, "Score-based generative modelling through stochastic differential equations", ICLR, 2021.

Prior: Approximating the score

Knowing the score function enables sampling from the prior. Approximate it instead:

$$\begin{aligned} \mathrm{d}\mathbf{s}_t &= [\mathbf{f}(\mathbf{s}_t) - g(t)^2 \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\ &\approx [\mathbf{f}(\mathbf{s}_t) - g(t)^2 \mathbf{S}_{\theta^*}(\mathbf{s}_t, t)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \end{aligned}$$

Prior: Approximating the score

Knowing the score function enables sampling from the prior. Approximate it instead:

$$\begin{aligned} \mathrm{d}\mathbf{s}_t &= [\mathbf{f}(\mathbf{s}_t) - g(t)^2 \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\ &\approx [\mathbf{f}(\mathbf{s}_t) - g(t)^2 \mathbf{S}_{\theta^*}(\mathbf{s}_t, t)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \end{aligned}$$

1. Learn $\mathbf{S}_\theta(\mathbf{s}_t, t)$:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{t,\mathbf{s},\zeta,\mathbf{s}_t|\mathbf{s}}\left[\|\mathbf{S}_\theta(\mathbf{s}_t, t) + \frac{\zeta}{\sigma(t)}\|_2^2\right], \quad \zeta \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I})$$

Prior: Approximating the score

Knowing the score function enables sampling from the prior. Approximate it instead:

$$\mathrm{d}\mathbf{s}_t = [\mathbf{f}(\mathbf{s}_t) - g(t)^2 \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$
$$\approx [\mathbf{f}(\mathbf{s}_t) - g(t)^2 \mathbf{S}_{\theta^*}(\mathbf{s}_t, t)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

1. Learn $\mathbf{S}_\theta(\mathbf{s}_t, t)$:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{t,\mathbf{s},\zeta,\mathbf{s}_t|\mathbf{s}}\left[\left\|\mathbf{S}_\theta(\mathbf{s}_t, t) + \frac{\zeta}{\sigma(t)}\right\|_2^2\right], \quad \zeta \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I})$$

2. Numerically sample from the prior $p_\theta(\mathbf{s})$

☞ The above SDE can be solved by the *Predictor-Corrector (PC) sampler*

SE phase as EM approach

Once the prior *score model* is trained, SE is performed via EM:

E-step: Approximate the conditional reverse SDE:

$$\mathrm{d}\mathbf{s}_t = \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t|\mathbf{x})\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

[3] X. Meng and Y. Kabashima, "Diffusion model based posterior sampling for noisy linear inverse problems," 2022.

SE phase as EM approach

Once the prior *score model* is trained, SE is performed via EM:

E-step: Approximate the conditional reverse SDE:

$$\begin{aligned}\mathrm{d}\mathbf{s}_t &= \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t|\mathbf{x})\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\ &= \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \left(\nabla_{\mathbf{s}_t} \log p_\phi(\mathbf{x}|\mathbf{s}_t) + \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t) \right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}\end{aligned}$$

[3] X. Meng and Y. Kabashima, "Diffusion model based posterior sampling for noisy linear inverse problems," 2022.

SE phase as EM approach

Once the prior *score model* is trained, SE is performed via EM:

E-step: Approximate the conditional reverse SDE:

$$\begin{aligned}
\mathrm{d}\mathbf{s}_t &= \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t|\mathbf{x})\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\
&= \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \left(\nabla_{\mathbf{s}_t} \log p_\phi(\mathbf{x}|\mathbf{s}_t) + \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t)\right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}
\end{aligned}$$

⚠ *Intractable*, time-dependent likelihood!

Approximation by the *"noise-perturbed pseudo-likelihood score"*[3] $\nabla_{\mathbf{s}_t} \log \tilde{p}_\phi(\mathbf{x}|\mathbf{s}_t)$

[3] X. Meng and Y. Kabashima, "Diffusion model based posterior sampling for noisy linear inverse problems," 2022.

SE phase as EM approach

Once the prior *score model* is trained, SE is performed via EM:

E-step: Approximate the conditional reverse SDE:

$$
\begin{aligned}
\mathrm{d}\mathbf{s}_t &= \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t|\mathbf{x})\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\
&= \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \left(\nabla_{\mathbf{s}_t} \log p_\phi(\mathbf{x}|\mathbf{s}_t) + \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t)\right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}
\end{aligned}
$$

⚠ *Intractable*, time-dependent likelihood!

Approximation by the *"noise-perturbed pseudo-likelihood score"*[3] $\nabla_{\mathbf{s}_t} \log \tilde{p}_\phi(\mathbf{x}|\mathbf{s}_t)$

$$
\tilde{p}_\phi(\mathbf{x}|\mathbf{s}_t) \sim \mathcal{N}_\mathbb{C}\left(\frac{\mathbf{s}_t}{\delta_t}, \frac{\sigma(t)^2}{\delta_t^2}\mathbf{I} + \mathrm{diag}(\boldsymbol{v}_\phi)\right), \qquad \delta_t = \mathrm{e}^{-\gamma t}
$$

[3] X. Meng and Y. Kabashima, "Diffusion model based posterior sampling for noisy linear inverse problems," 2022.

SE phase as EM approach

Once the *prior score model* is trained, SE is performed via EM:

E-step: Approximate the conditional reverse SDE:

$$
\begin{aligned}
\mathrm{d}\mathbf{s}_t &= \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t|\mathbf{x})\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\
&= \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \left(\nabla_{\mathbf{s}_t} \log p_\phi(\mathbf{x}|\mathbf{s}_t) + \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t) \right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\
&\approx \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \left(\lambda \nabla_{\mathbf{s}_t} \log \tilde{p}_\phi(\mathbf{x}|\mathbf{s}_t) + \mathbf{S}_{\theta^*}(\mathbf{s}_t, t) \right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}
\end{aligned}
$$

SE phase as EM approach

Once the *prior score model* is trained, SE is performed via EM:

E-step: Approximate the conditional reverse SDE:

$$\begin{aligned} \mathrm{d}\mathbf{s}_t &= \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t|\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\ &= \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \left(\nabla_{\mathbf{s}_t} \log p_\phi(\mathbf{x}|\mathbf{s}_t) + \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t)\right)\right] \mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\ &\approx \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \left(\lambda \nabla_{\mathbf{s}_t} \log \tilde{p}_\phi(\mathbf{x}|\mathbf{s}_t) + \mathbf{S}_{\theta^*}(\mathbf{s}_t, t)\right)\right] \mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \end{aligned}$$

- λ: weighting parameter to balance prior and likelihood terms.

SE phase as EM approach

E-step:

$$\mathrm{d}\mathbf{s}_t = \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2\left(\lambda\nabla_{\mathbf{s}_t}\log\tilde{p}_\phi(\mathbf{x}|\mathbf{s}_t) + \mathbf{S}_{\theta^*}(\mathbf{s}_t, t)\right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

SE phase as EM approach

E-step:

$$\mathrm{d}\mathbf{s}_t = \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2 \left(\lambda \nabla_{\mathbf{s}_t} \log \tilde{p}_\phi(\mathbf{x}|\mathbf{s}_t) + \mathbf{S}_{\theta^*}(\mathbf{s}_t, t)\right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

M-step:

SE phase as EM approach

E-step:

$$\mathrm{d}\mathbf{s}_t = \left[\mathbf{f}(\mathbf{s}_t) - g(t)^2\left(\lambda\nabla_{\mathbf{s}_t} \log \tilde{p}_\phi(\mathbf{x}|\mathbf{s}_t) + \mathbf{S}_{\theta^*}(\mathbf{s}_t, t)\right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

M-step:

$$\phi^* \leftarrow \underset{\mathbf{v}_\phi(i)\geq 0}{\operatorname{argmax}} \; \log p_\phi(\mathbf{x}|\hat{\mathbf{s}})$$

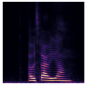
$$= \underset{\mathbf{v}_\phi(i)\geq 0}{\operatorname{argmin}} \; \sum_i \frac{(\mathbf{x}-\hat{\mathbf{s}})_i^*(\mathbf{x}-\hat{\mathbf{s}})_i}{\mathbf{v}_\phi(i)} + \log(\mathbf{v}_\phi(i))$$

UDiffSE pipeline

Pre-training		Posterior sampling		Output
$p_\theta(\mathbf{s})$	$\cdot$	$p_\phi(\mathbf{x}\|\mathbf{s})$	$\propto$	$p_\Theta(\mathbf{s}\|\mathbf{x})$

Clean speech estimate
(posterior sample)

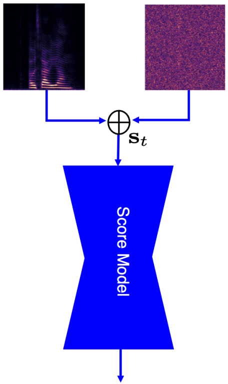
$\hat{\mathbf{s}}$



UDiffSE pipeline

Pre-training		Posterior sampling		Output
$p_\theta(\mathbf{s})$	$\cdot$	$p_\phi(\mathbf{x}\|\mathbf{s})$	$\propto$	$p_\Theta(\mathbf{s}\|\mathbf{x})$

$\mathbf{s}$ $\zeta \sim \mathcal{N}_\mathbb{C}(\mathbf{0}, \mathbf{I})$

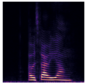


$\mathbf{s}_t$

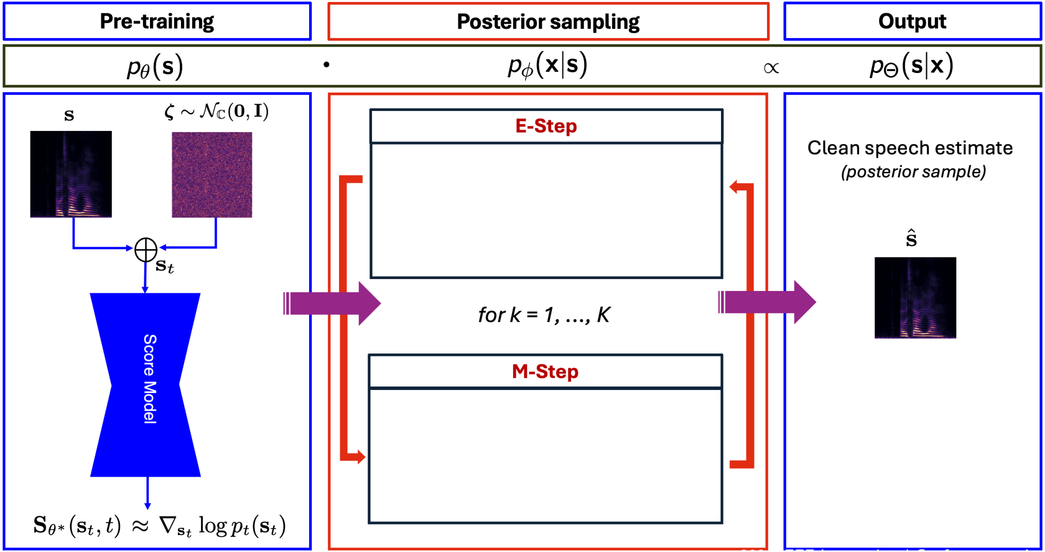
$$\mathbf{S}_{\theta^*}(\mathbf{s}_t, t) \approx \nabla_{\mathbf{s}_t} \log p_t(\mathbf{s}_t)$$

Clean speech estimate
(posterior sample)

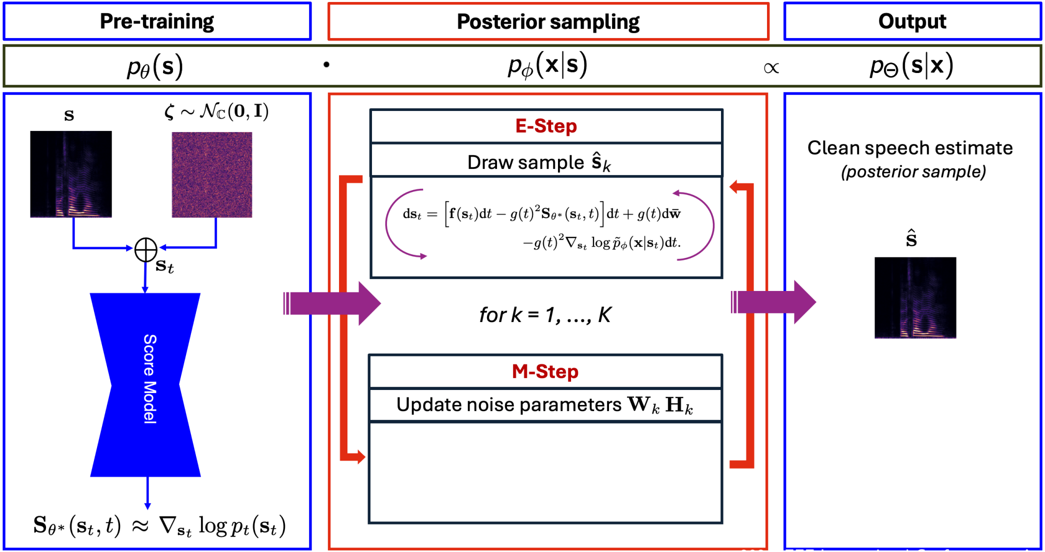
$\hat{\mathbf{s}}$



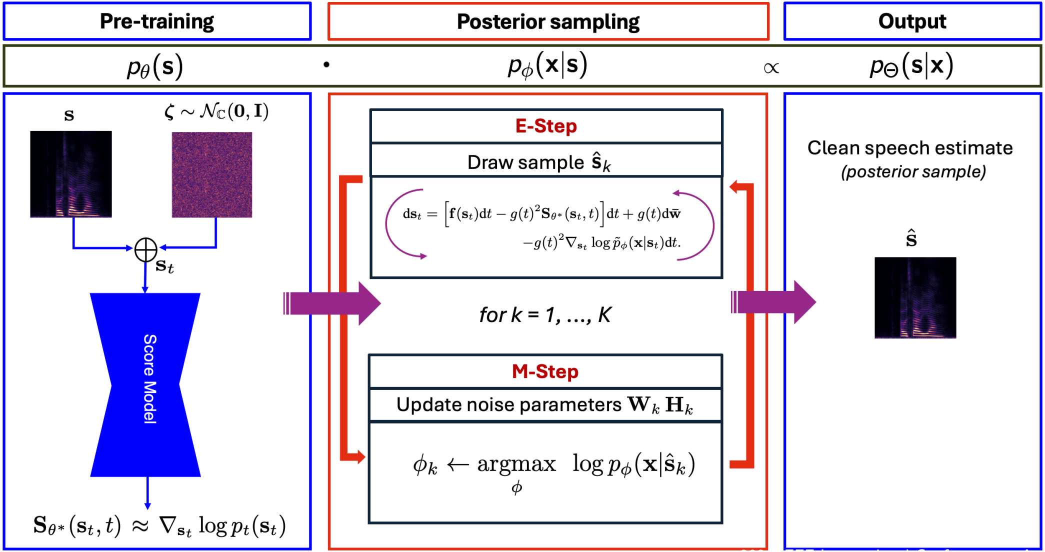
UDiffSE pipeline



UDiffSE pipeline



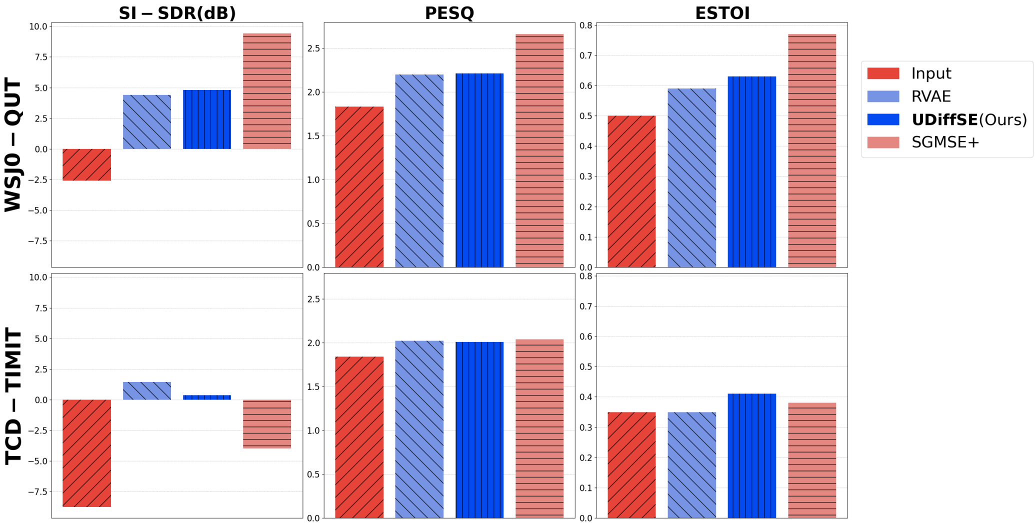
UDiffSE pipeline



Experiments

- **Datasets**.
  - Training: WSJ0 ($\sim$ 25hrs)
  - Testing: WSJ0-QUT (1.5hrs), TCD-TIMIT (45mins)
  - Noise levels (dB): $[-5, 0, 5]$.
  - Noise types: *Café*, *Home*, *Street*, and *Car*
- **Evaluation Metrics**.
  - Objective measures: SI-SDR, ESTOI, PESQ
  - (Pseudo)-subjective measures: DNS-MOS (SIG, BAK, OVRL)
- **Baselines**. RVAE, SGMSE+ (pre-trained).
- **Models architecture**. Multi-resolution U-Net as in SGMSE+.
- **EM settings**. NMF rank 4. $K = 5$ EM iterations. Averaging over $b = 4$ parallel sample batches. Weighting parameter $\lambda = 1.5$.

Results


SI – SDR(dB)
PESQ
ESTOI
WSJ0 – QUT
TCD – TIMIT
Input
RVAE
UDiffSE(Ours)
SGMSE+

Conclusion & next directions

▷ **Conclusions**

- UDiffSE: *Proof of concept*
- Learning an *implicit prior* distribution over clean speech data
- An EM approach to generate clean speech & learn the noise parameters *at the same time*
- Better *generalisation* & outperforms VAE (also less artifacts)

Conclusion & next directions

- **Conclusions**
  - UDiffSE: *Proof of concept*
  - Learning an *implicit prior* distribution over clean speech data
  - An EM approach to generate clean speech & learn the noise parameters *at the same time*
  - Better *generalisation* & outperforms VAE (also less artifacts)
- **Next steps**
  1. Speeding up inference
  2. Investigating generalisational capability
  3. Improving prior

Further resources

GitHub

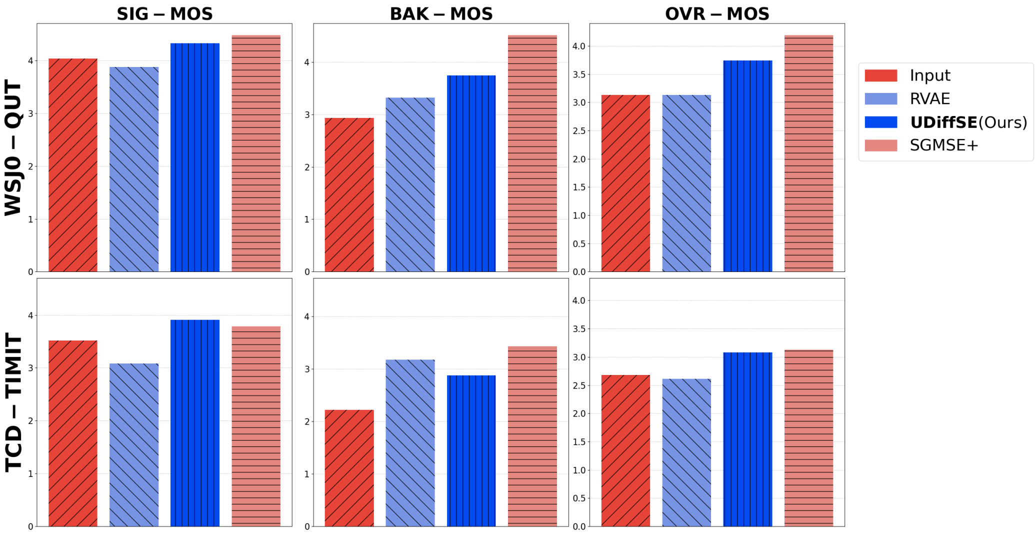
https://github.com/joanne-b-nortier/udiffse

Demo

https://team.inria.fr/multispeech/demos/udiffse/

Additional resources

Results II


SIG – MOS
BAK – MOS
OVR – MOS
WSJ0 – QUT
TCD – TIMIT
Input
RVAE
UDiffSE(Ours)
SGMSE+

Algorithm

Algorithm 2 Posterior sampling (E-step) of UDiffSE

Require: $\mathbf{x}, N, \ell, \lambda, r$ (signal-to-noise ratio)

$\mathbf{s}_1 \sim \mathcal{N}_{\mathbb{C}}(\mathbf{x}, \mathbf{I}), \Delta\tau \leftarrow \frac{1}{N}$

for $i = N, \ldots, 1$ **do**

  $\tau \leftarrow \frac{i}{N}$

  $\epsilon_\tau \leftarrow (\sigma_\tau \cdot r)^2$

  $\boldsymbol{\zeta}_c \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I})$ ▷ *(Corrector)*

  $\mathbf{s}_\tau \leftarrow \mathbf{s}_\tau + \epsilon_\tau \mathbf{S}_{\theta^*}(\mathbf{s}_\tau, \tau) + \sqrt{2\epsilon_\tau}\boldsymbol{\zeta}_c$

  $\boldsymbol{\zeta}_p \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I})$ ▷ *(Predictor)*

  $\mathbf{s}_\tau \leftarrow \mathbf{s}_\tau - \mathbf{f}_\tau \Delta\tau + g_\tau^2 \mathbf{S}_{\theta^*}(\mathbf{s}_\tau, \tau)\Delta\tau + g_\tau \sqrt{\Delta\tau}\boldsymbol{\zeta}_p$

  if $i \equiv 0 \pmod{\ell}$ **then** ▷ *(Posterior)*

    $\nabla_{\mathbf{s}_\tau} \log \tilde{p}_\phi(\mathbf{x}|\mathbf{s}_\tau) \leftarrow \frac{1}{\delta_\tau}\left[\frac{\sigma_\tau^2}{\delta_\tau^2}\mathbf{I} + \mathrm{diag}(\boldsymbol{v}_\phi)\right]^{-1}\left(\frac{\mathbf{s}_\tau}{\delta_\tau} - \mathbf{x}\right)$

    $\mathbf{s}_\tau \leftarrow \mathbf{s}_\tau + \lambda g_\tau^2 \nabla_{\mathbf{s}_\tau} \log \tilde{p}_\phi(\mathbf{x}|\mathbf{s}_\tau)\Delta\tau$

  end if

end for

return $\hat{\mathbf{s}} = \mathbf{s}_0$