Disentangling Noise from Images: A Flow-Based Image Denoising Neural Network

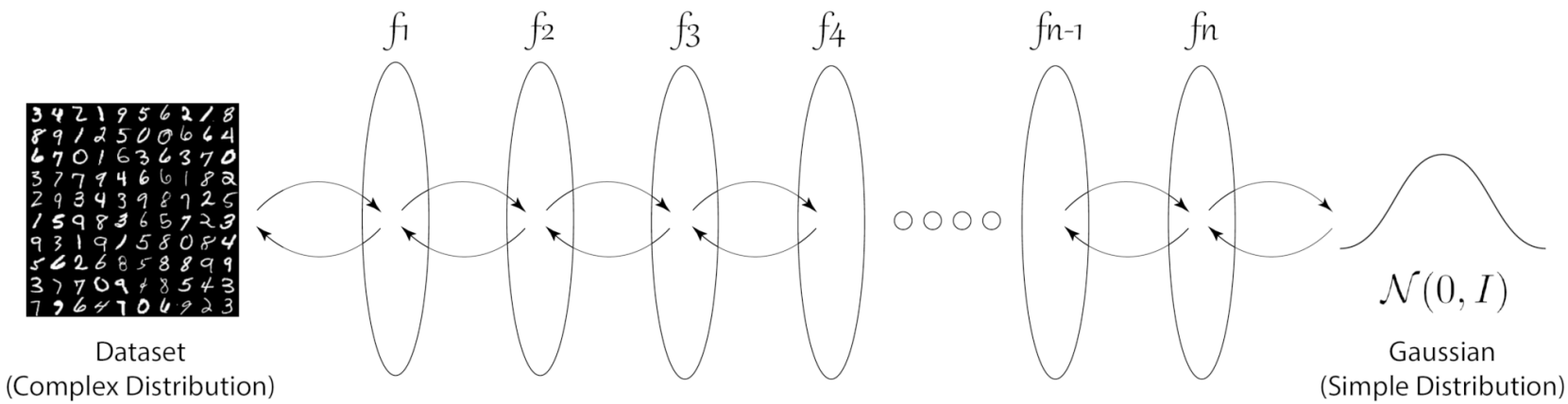
Yang Liu et al.,
https://www.mdpi.com/1424-8220/22/24/9844

Short Notes

February 2026

Reminder: The key idea of the Normalizing Flows

- **simple** $f \Rightarrow$ **small** expressivity
- **complex** $f \Rightarrow$ **hard** calculations
- $\Rightarrow$ **key** idea:
  - **chain** of simple $1 \leftrightarrow 1$ invertible transformations = **Normalizing Flow**
  - network learning $\Rightarrow$ **log likelihood**



Reminder:Models with NF: RealNVP

affine coupling layer f: $\mathbf{x} \mapsto \mathbf{y}$:

$$\begin{aligned} \mathbf{y}_{1:d} &= \mathbf{x}_{1:d} \\ \mathbf{y}_{d+1:D} &= \mathbf{x}_{d+1:D} \odot \exp(\sigma(\mathbf{x}_{1:d})) + \mu(\mathbf{x}_{1:d}) \end{aligned} \tag{1}$$

where $\sigma(.)$ and $\mu(.)$ are scale and translation functions and both map $\mathbb{R}^d \mapsto \mathbb{R}^{D-d}$. The operation $\odot$ is the element-wise product.

- easily invertible + to calculate the Jacobian

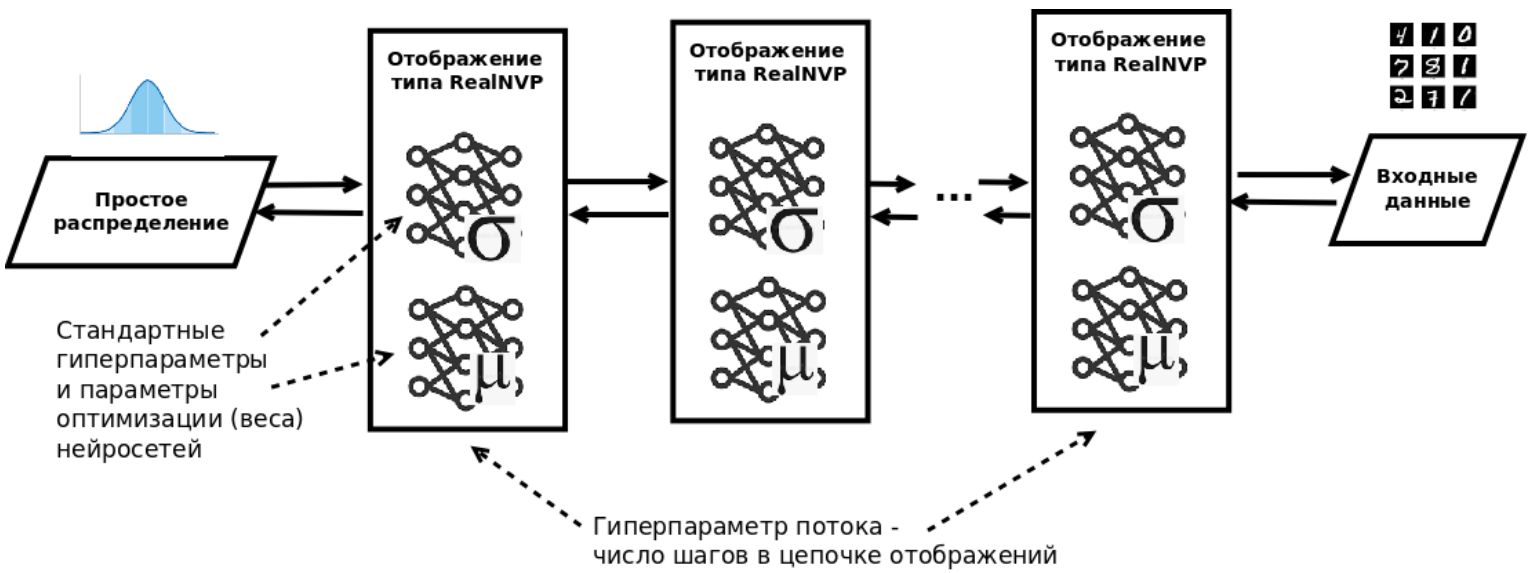
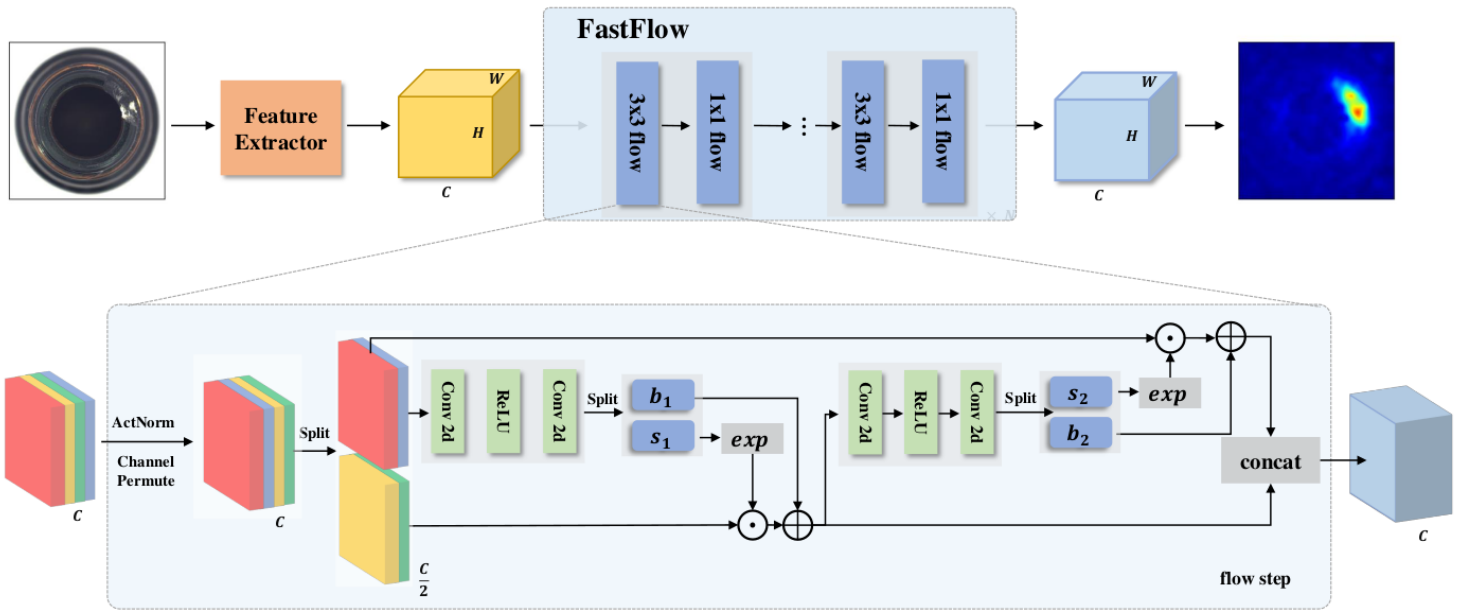


Рис.: Блок-схема нормализующих потоков для моделей типа RealNVP

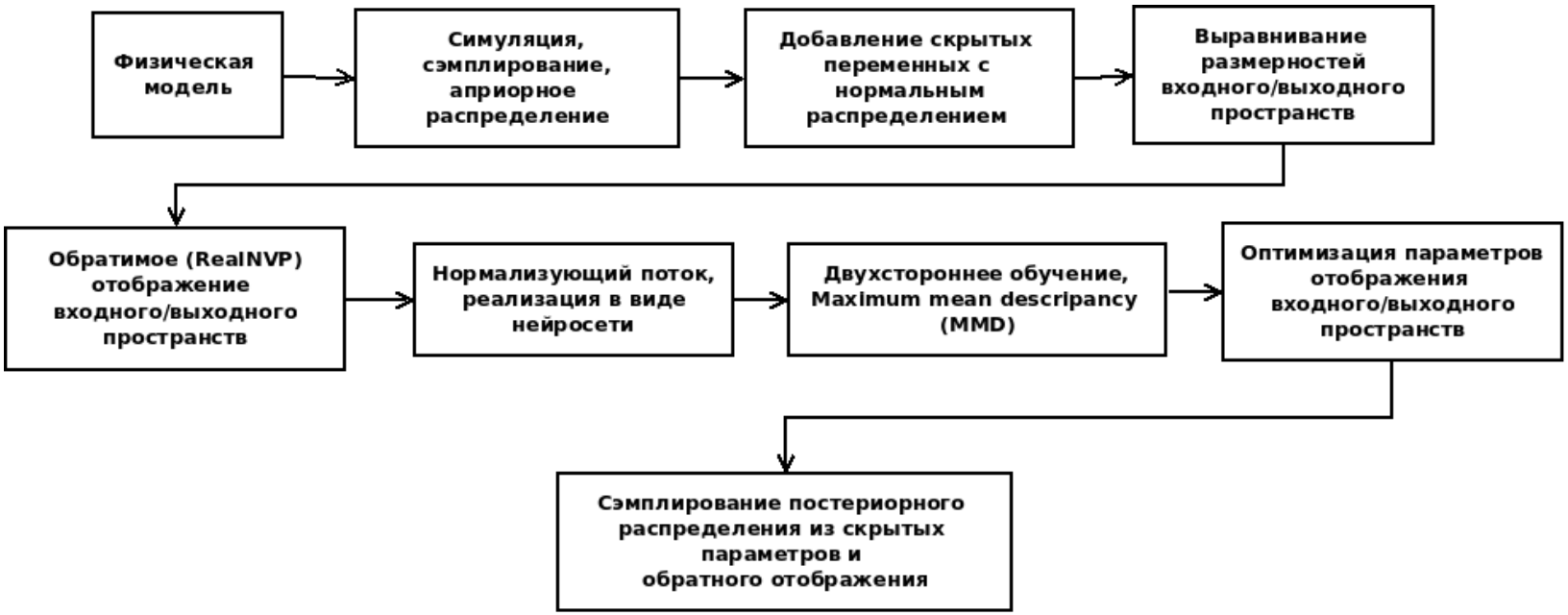
Reminder: алгоритм детектирования аномалий

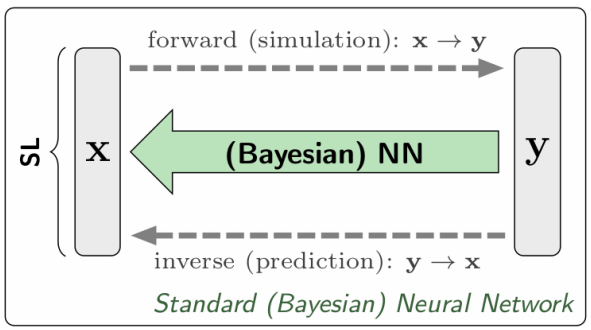
1. поскольку отображение (NF)оптимизируется под выборку **неаномальных** данных, аномалии имеют низкую вероятность
   - в исходном пространстве со сложным распределением (+имеем только выборку) сложно провести границу между нормальными и аномальными данными
   - для нормального распределения – легко, например, граница определяется 2σ (95%)

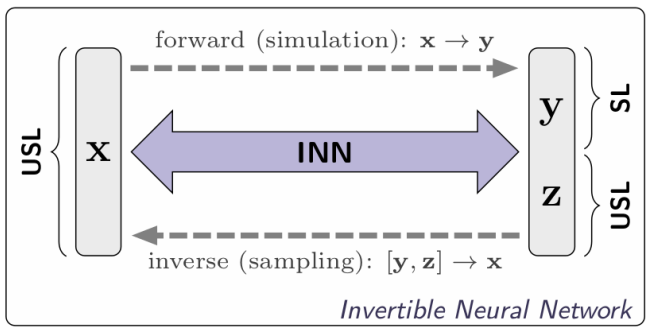


Reminder: Inverse Problems with Invertible Neural Networks

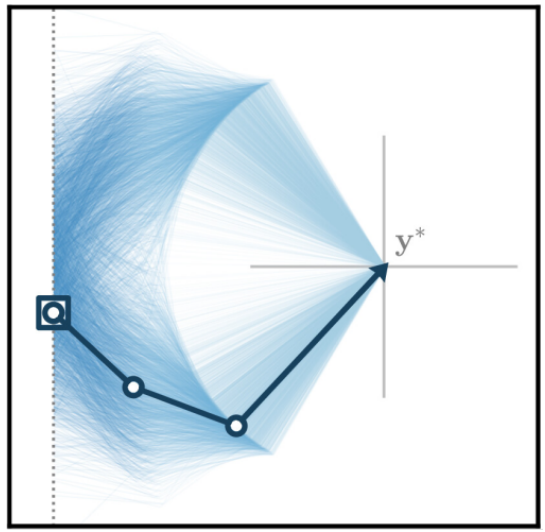
- model $y = s(x)$ for the forward process
- aim at approximating $p(\mathbf{x} \mid \mathbf{y})$

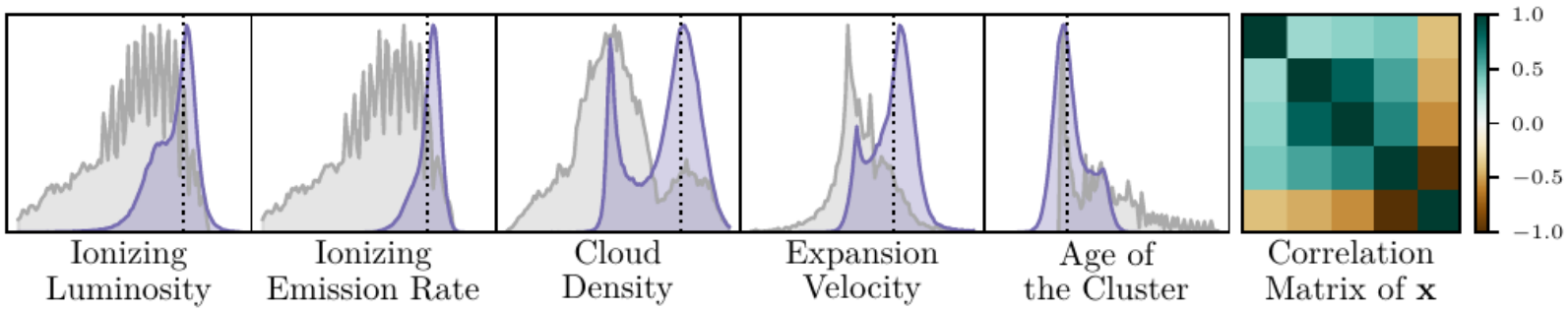




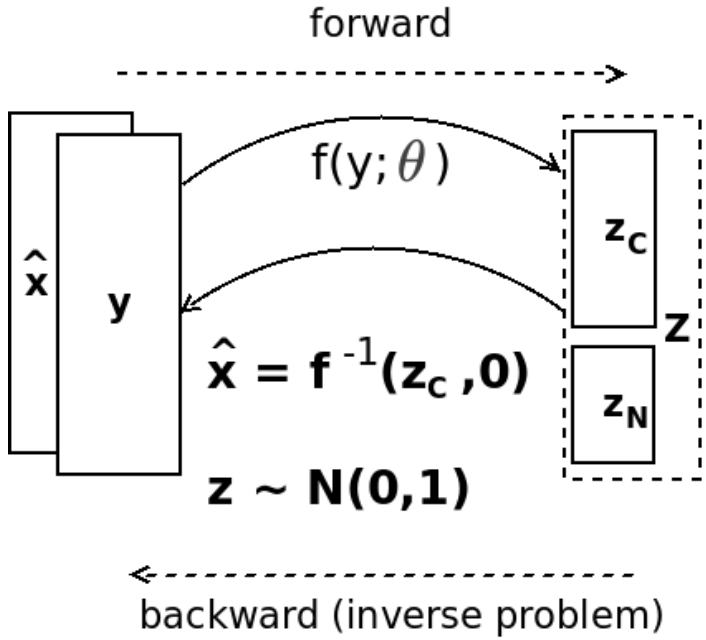


Reminder: Inverse Problems with INN (2)





INN for image denoising

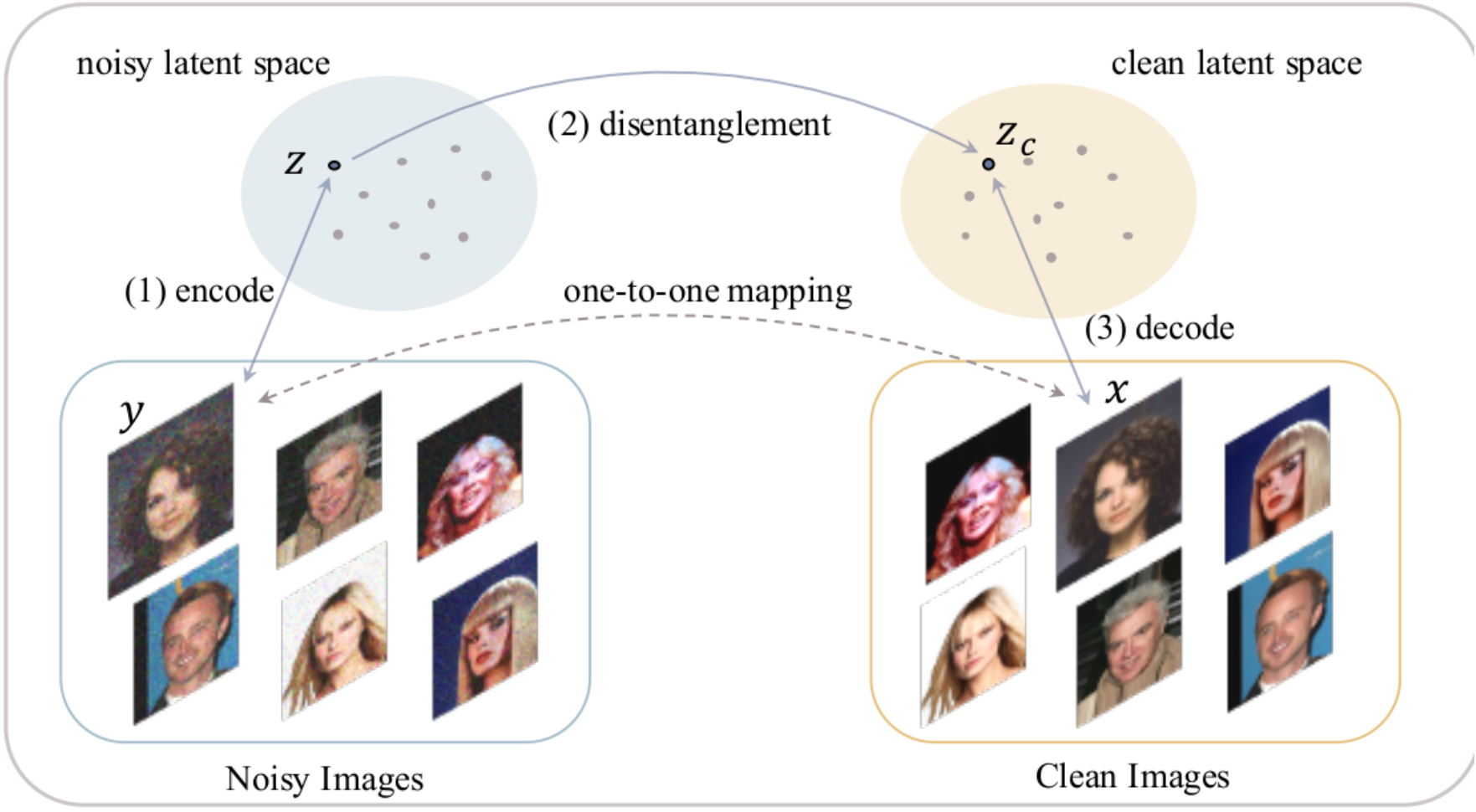


- Есть пары данных $\{\mathbf{x}, \mathbf{y}\}$, где
  - $\mathbf{x}$ — чистое изображение,
  - $\mathbf{y} = \mathbf{x} + \eta$ — **синтетически** зашумлённое изображение;
  - η — шум (может быть гауссовским, реалистичным, и т.д.).
- Цель INN: построить обратимое отображение f_θ, такое что при прямом проходе (forward) $f_\theta(\mathbf{y}) = \mathbf{z} \equiv [\mathbf{z}_C, \mathbf{z}_N]$
  - где $\mathbf{z}_C$ — латент, кодирующий "чистую" структуру / содержимое,
  - $\mathbf{z}_N$ — латент, содержащий шум/неопределённость.

Что хотим получить

- В обратном направлении (inverse) хотим получить чистое изображение, подавив/заменив шумовую составляющую: $\hat{\mathbf{x}} = f_\theta^{-1}(\mathbf{z}_C, \mathbf{z}'_N)$
- обычно подставляют $\mathbf{z}'_N = 0$ или $\mathbf{z}'_N \sim p(\mathbf{z}_N)$ (если хотим стохастичность)(?).
- Таким образом, данная схема включает три шага:
  1. изучение распределения зашумленных изображений путем кодирования $\mathbf{y}$ в зашумленное скрытое представление $\mathbf{z}$,
  2. отделение чистого представления $\mathbf{z}_C$ от $\mathbf{z}$,
  3. восстановление чистого изображения путем декодирования $\mathbf{z}_C$ в пространство чистого изображения.
- Для обеспечения эффекта шумоподавления отображения между $\mathbf{y}$ и $\mathbf{z}$, $\mathbf{z}_C$ и $\mathbf{x}$ должны быть взаимно однозначными.

Три шага схемы шумоподавления



Изображения $\leftrightarrow$ латенты

$$p(\mathbf{y}) = p(\mathbf{z}) \left| \det \frac{df(\mathbf{z})}{d\mathbf{z}} \right|^{-1}$$

- assume that $\mathbf{z}$ can be disentangled:
  - some of the dimensions of $\mathbf{z}$ encode the distribution of clean images (denoted as $\mathbf{z}_C$) and the remaining embed noise (denoted as $\mathbf{z}_N$).
- The clean image x can be restored through the following transformation:

$$p(\mathbf{x}) = p(\mathbf{z_C}) \left| \det \frac{df(\mathbf{z_C})}{d\mathbf{z}} \right|^{-1}$$

- However, how to obtain $\mathbf{z}_C$ with $\mathbf{z}$ "is not so obvious".
- It is proposed a way of disentanglement by setting $\mathbf{z}_N = 0$

Обучение: функции потерь (1)

- Реконструкционная потеря (supervised):
  $\mathcal{L}_{\text{rec}} = ||\hat{\mathbf{x}} - \mathbf{x}||_1 \quad \text{или} \quad ||\hat{\mathbf{x}} - \mathbf{x}||_2^2$
  - Это обеспечивает, что $\mathbf{z}_C$ являются существенными параметрами для чистых изображений
- Ошибка "шумовой" латенты (matching noise):
  $\mathcal{L}_{\text{noise}} = ||\mathbf{z}_N - \boldsymbol{\eta}||_2^2$
  - если во время прямого прохода мы отдельно извлекаем $\mathbf{z}_N$ и в синтетическом наборе знаем шум $\boldsymbol{\eta}$, то их можно принудительно согласовать
    - как? они разных размерностей! : $\boldsymbol{\eta}$ - сложение, $\mathbf{u}$ - конкатенация

Обучение: функции потерь (2)

- Лог-правдоподобие: Flow-модель даёт явную вероятность латентов. Если модель такова, что $f_\theta(\mathbf{y}) = (\mathbf{z}, \mathbf{u})$, и мы предполагаем $p(\mathbf{u})$ (например, $\mathcal{N}(0, I)$), то отрицательное лог-правдоподобие для $\mathbf{y}$ превращается в:

$$\mathcal{L}_{\text{NLL}}(\mathbf{y}) = -\log p_z(\mathbf{z}) - \log \left| \det \frac{\partial f_\theta(\mathbf{y})}{\partial y} \right|$$

При условии факторизации $p_z(\mathbf{z}_C, \mathbf{z}_N) = p_z(\mathbf{z}_C) p_z(\mathbf{z}_N)$ иногда упрощают: минимизируем $-\log p_z(\mathbf{z}_N) - \log |\det J|$ (часть, связанная с $\mathbf{z}_C$, можно оставить без явной цели, если основной фокус на $\mathbf{z}_N$. Также на практике используют Gaussian prior для $\mathbf{z}_N \to (-\log p_z(\mathbf{z}_N) \propto |\mathbf{z}_N|^2)$.

Итоговая комбинированная функция потерь

- В общем случае: $\mathcal{L} = \alpha\mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{noise}} + \gamma\mathcal{L}_{\text{NLL}} + \delta\mathcal{L}_{\text{aux}}$
- где $\mathcal{L}_{\text{aux}}$ — дополнительные члены (SSIM loss, perceptual loss, adversarial loss и т.д.).
- Для стабильного обучения обычно α доминирует.
- Главное в обучении — контролировать, чтобы шум оказался в $\mathbf{z}_N$, а не "прятался" в $\mathbf{z}_C$.

Архитектура сети

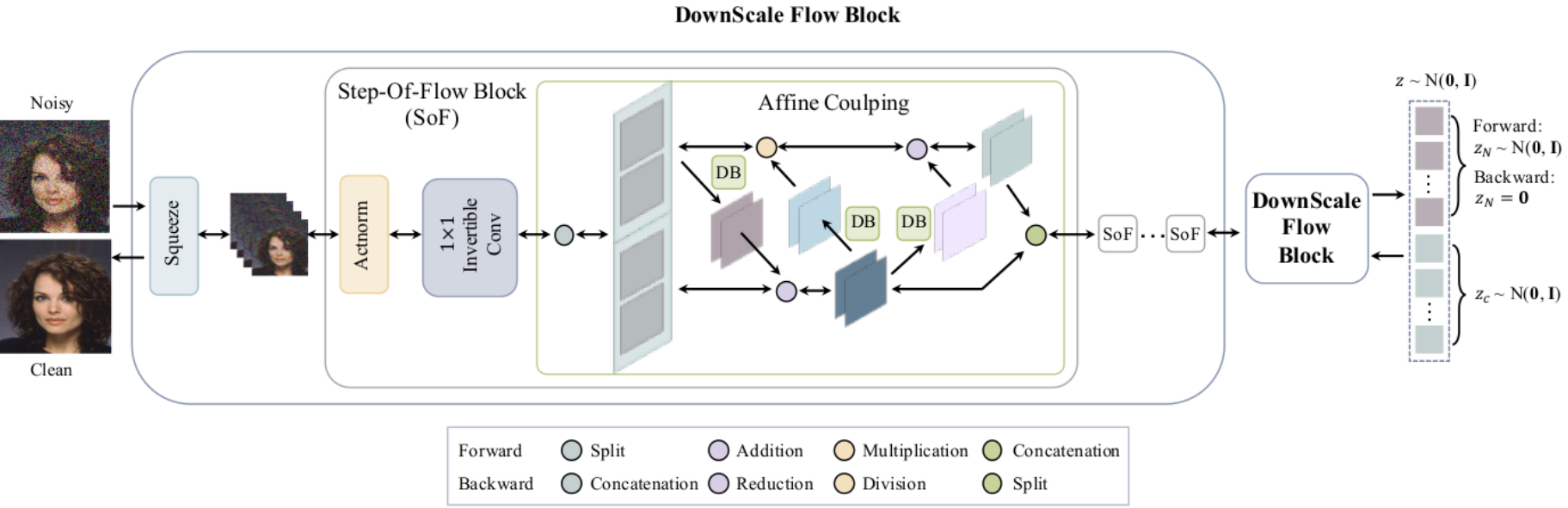


Table 4. Comparisons on the denoising accuracy of different numbers of DownScale Blocks and Invertible Blocks.

PSNR		SoF Blocks	
		Num = 4	Num = 8
Flow Blocks	num = 1	29.59	29.86
	num = 2	29.87	**30.00**
	num = 3	29.89	29.90

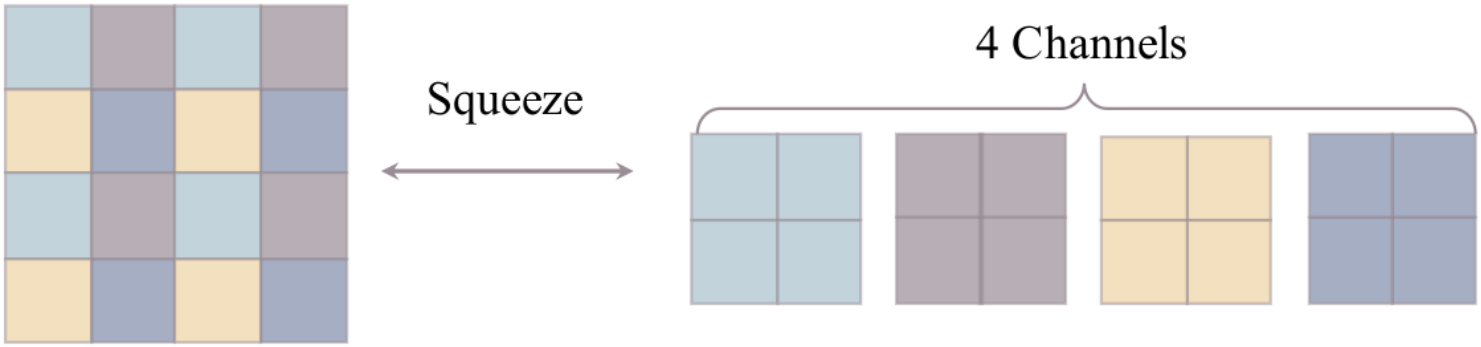
Выбор размерностей латентов

Table 5. Comparisons on different proportions of the dimensions of $\mathbf{z}_C$ in $\mathbf{z}$.

$\dim(\mathbf{z}_C)$	1/8	1/4	1/2	3/4	7/8
PSNR	29.81	30.00	30.00	**30.19**	30.18

- Выбирают:
  - 75% — content
  - 25% — noise
- Почему это работает
  - Визуальная структура требует больше степеней свободы, чем шум
  - Шум статистически проще (часто близок к Gaussian)
- Типичные значения из статей:
  - 10–30% каналов $\rightarrow$ noise
  - 70–90% каналов $\rightarrow$ content
- В INN-денойзинге размерность шума — это не параметр данных, а **гиперпараметр** модели, который задаёт где модель будет хранить неопределённость.

Слои: Squeeze



- Squeeze очень важен, без него подавление шума практически не работает.
- это детерминированная, полностью обратимая перестановка элементов тензора.
  - Никакая информация не теряется
  - Нет параметров
  - log-детерминант Якобиана $= 0$
- Coupling layers видят только по каналам
  - Без squeeze каждый пиксель изолирован, а локальные корреляции (2×2) не видны

Слои: Squeeze не равен pooling или stride

Операция	Обратима	Потеря информации	Назначение
MaxPool	❌	да	downsampling
AvgPool	❌	да	сглаживание
Stride Conv	❌	частично	feature extraction
Squeeze	✅	**нет**	**перераспределение информации**

➡️ Squeeze **меняет представление**, а не данные.

Слои: Activation Normalization (Actnorm)

- Actnorm replaces Batch Normalization,
  - performs an affine transformation with a scale and bias parameter per channel of the type

$$y = \frac{x - \mu}{\sigma}\gamma + \beta$$

    where γ and β are learned, initialized from the initial data statistics
  - Unlike Batch Normalization, which uses batch statistics throughout training, Actnorm calculates the mean and standard deviation from the first initial minibatch of data.
  - After the initial data-dependent setup, the scaling and shifting parameters become regular trainable parameters, independent of the input data.
  - allows for deeper models

Слои: 1×1 convolution

- Dimensionality Reduction/Increase: It acts as a bottleneck layer, reducing the number of channels (depth) to lower computational costs (e.g., in ResNet or Inception modules).
- It blends information across input channels for each spatial location.
- Non-Linearity: Similar to a fully connected layer applied per-pixel, it adds more non-linear activation functions (like ReLU) to the network, increasing capacity without significantly increasing parameters.
- Feature Combination: It creates a new set of features by learning optimal linear combinations of existing input channels.

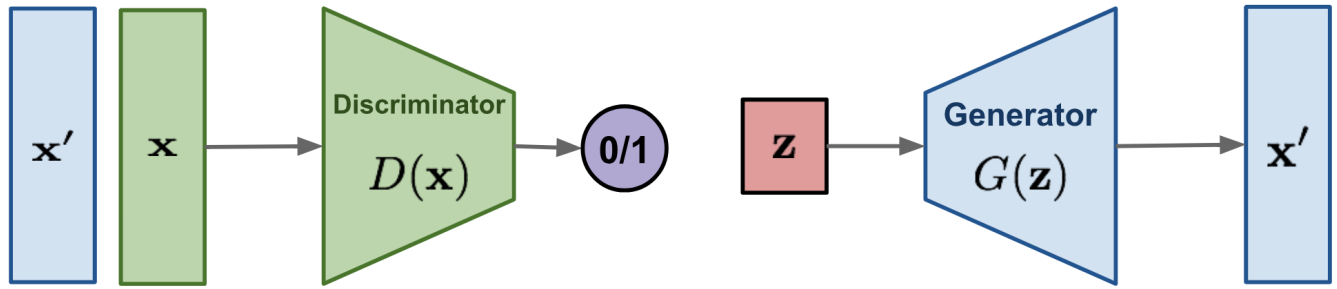
$$Y_d(i,j) = \sum_{c=1}^{C} \sum_{m} \sum_{n} X_c(i+m, j+n) \cdot K_{d,c}(m,n) + b_d$$

Еще литература

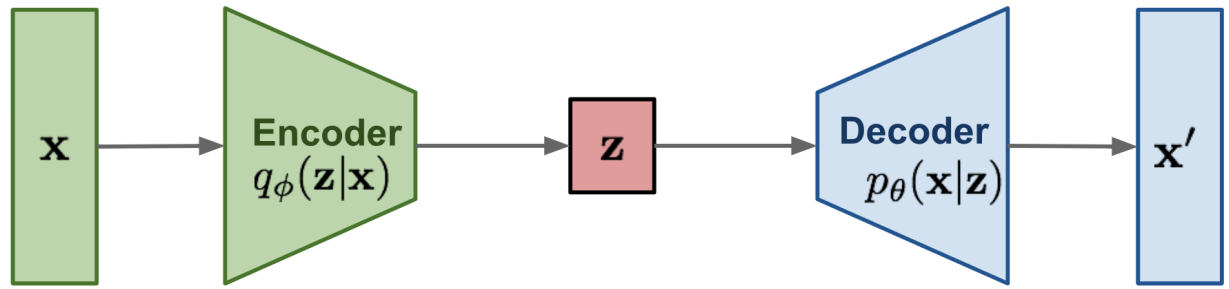
- Abdelhamed, A., ert al "Noise flow: Noise modeling with conditional normalizing flows." http://openaccess.thecvf.com/content_ICCV_2019/papers/Abdelhamed_Noise_Flow_Noise_Modeling_With_Conditional_Normalizing_Flows_ICCV_2019_paper.pdf
- Lugmayr, A, et al "Srflow: Learning the super-resolution space with normalizing flow." https://arxiv.org/pdf/2006.14200
- Asim, M. et al, "Invertible generative models for inverse problems: mitigating representation error and dataset bias." http://proceedings.mlr.press/v119/asim20a/asim20a.pdf

Types of Generative Models

GAN: minimax the classification error loss.



VAE: maximize ELBO.



Flow-based generative models: minimize the negative log-likelihood