

Sequence Models, Attention and Generative Models

Pattern Recognition and Machine Learning



Lecture 18 Sequence Models, Attention and Generative Models

Chizhi Chris ZHANG

zhangchizhi@ciomp.ac.cn

Advanced Computing and Digital Technology Research Center

University of Chinese Academy of Sciences

Autumn 2026

Lecture plan and time allocation

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Topic	Minutes
Recurrent models	45
Attention and Transformers	45
Generative models	45

Information access and training targets

Follow sequence memory; calculate attention; compare how new samples are generated.

The numerical steps are shown on the slides.



Autoregressive likelihood makes the prediction task explicit

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The current input may not contain enough information to predict the target.

A feedforward model needs a fixed input representation; sequence lengths and relevant distances vary.

The model returns a probability distribution over the vocabulary at each position.

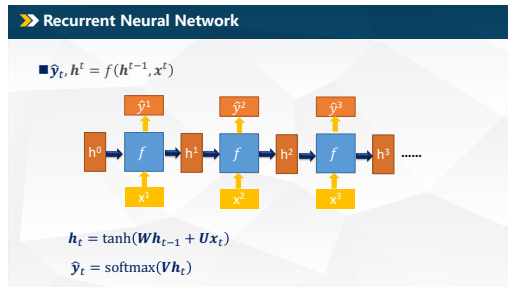
$$p(x_1, \dots, x_T) = \prod_{t=1}^T p(x_t \mid x_{<t}), \quad L = - \sum_{t=1}^T \log p_{\theta}(x_t \mid x_{<t}).$$



An RNN reuses one transition at every time step

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Input x_t and previous state h_{t-1} produce h_t . The same transition function is used again at $t + 1$.



A scalar recurrence shows how old inputs survive

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Key idea

It is a learned internal representation. After reading x_t , p_t can predict the next token x_{t+1} .

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b_h), \quad o_t = W_o h_t + b_o, \quad p_t = \text{softmax}(o_t).$$

The current state mixes previous inputs with distance-dependent weights. This linear example explains memory flow; an actual RNN uses learned matrices and nonlinear activations.

$$h_t = 0.5h_{t-1} + x_t \quad h_0 = 0$$

Time	Input	State calculation	State
1	1	$0.5(0) + 1$	1
2	2	$0.5(1) + 2$	2.5
3	0	$0.5(2.5) + 0$	1.25

$$h_3 = 0.5^2 x_1 + 0.5 x_2 + x_3 = 0.25 + 1 + 0 = 1.25.$$

The same inputs in a different order give a different state

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Both sequences contain the same numbers and have the same sum.

$$h_t = 0.5h_{t-1} + x_t \quad h_0 = 0$$

Input order	State sequence	Final state
(1, 2, 0)	1, 2.5, 1.25	1.25
(0, 2, 1)	0, 2, 2	2



Shared recurrence: add contributions from different uses

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Backpropagation through time is the chain rule applied to this expanded graph.

$$L = \sum_{t=1}^T \ell_t, \quad \frac{\partial L}{\partial W_h} = \sum_{t=1}^T \frac{\partial L}{\partial W_h} \Big|_{\text{use at } t}.$$

The two occurrences refer to one shared parameter a .

$$h_t = ah_{t-1} + x_t \quad h_0 = 0$$

$$(x_1, x_2, x_3) = (1, 2, 0) \quad h_1 = 1$$

$$h_2 = a + 2$$

$$h_3 = a(a + 2) = a^2 + 2a.$$

$$\frac{dh_3}{da} = 2a + 2 = 3 \quad \text{at } a = .5.$$

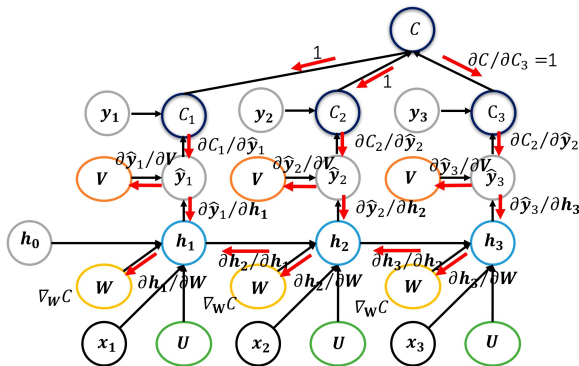
$$\frac{dh_3}{da} = \underbrace{h_2}_{\text{direct use at step 3}} + \underbrace{a \frac{dh_2}{da}}_{\text{through the earlier state}} = 2.5 + .5(1) = 3.$$



Repeated time steps share the same parameters

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Each column is one time step.
Repeated symbols U, W, V denote shared parameters. Forward paths compute states and losses; backward paths add gradient contributions from every use of the same parameter.



Clipping limits explosions; it cannot recover vanished gradients

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A signal important far in the past can be difficult to learn even when a short sequence works.

$$s < t$$

$$\frac{\partial h_t}{\partial h_s} = J_t J_{t-1} \cdots J_{s+1}, \quad J_k = \text{diag}(1 - h_k^2) W_h.$$

For $\mathbf{g} = (3, 4)$, $\|\mathbf{g}\|_2 = 5$. With $c = 1$, the clipped gradient is $(0.6, 0.8)$: same direction, norm one.

$$\mathbf{g}_{\text{clip}} = \mathbf{g} \min \left(1, \frac{c}{\|\mathbf{g}\|_2} \right).$$

$$\mathbf{g} = (3, 4) \quad \|\mathbf{g}\|_2 = 5$$

$$c = 1$$

$$\mathbf{g}_{\text{clip}} = (3, 4) \times \frac{1}{5} = (0.6, 0.8), \quad \|\mathbf{g}_{\text{clip}}\| = 1.$$

LSTM gates are computed from input and previous state

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Cell state c_t carries a memory path; hidden state h_t is what the unit exposes for subsequent computation.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t,$$

$$h_t = o_t \odot \tanh(c_t).$$

Its local partial derivative, holding gates fixed, is f_t . A gate near one can support retention, but LSTM does not guarantee unlimited memory.

$$u_t = [h_{t-1}; x_t]$$

$$f_t = \sigma(W_f u_t + b_f), \quad i_t = \sigma(W_i u_t + b_i),$$

$$o_t = \sigma(W_o u_t + b_o), \quad \tilde{c}_t = \tanh(W_c u_t + b_c),$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad h_t = o_t \odot \tanh(c_t).$$

LSTM: retain, write, then expose a numeric memory

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Given gates for one scalar update

$$c_{t-1} = 2, \quad f_t = .8,$$

$$i_t = .25, \quad \tilde{c}_t = -.4, \quad o_t = .6.$$

The forget gate retains old memory.

The input gate controls new content.

The output gate exposes a transformed state.

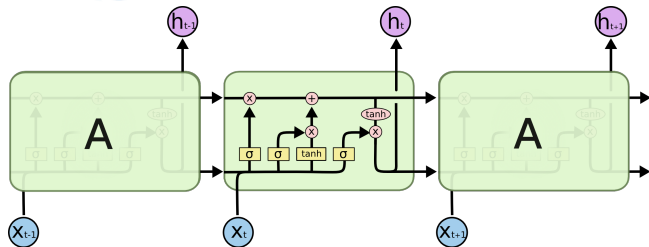
Memory update

Quantity	Calculation	Value
Retained	$.8(2)$	1.6
Written	$.25(-.4)$	-.1
c_t	$1.6 - .1$	1.5
h_t	$.6 \tanh(1.5)$	.543

c_t carries memory; h_t is the exposed output. Memory 1.5 is compatible with a bounded hidden output.

Follow the memory path in the original LSTM diagram

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER



The top path retains old memory and adds candidate content. The right branch exposes memory through tanh and an output gate. Adjacent cells share parameters.

$$c_t = 1.5$$



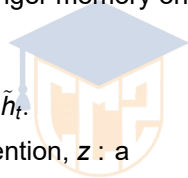
GRU is a compact alternative, not a guaranteed winner

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Compare suitable baselines at similar resources. Neither model guarantees stronger memory on every task.

$$\tilde{h}_t = \tanh(W_x x_t + U(r_t \odot h_{t-1}) + b), \quad h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t.$$

If $h_{t-1} = 2$, $\tilde{h}_t = -0.4$, $z_t = 0.25$, and $h_t = 0.75(2) + 0.25(-0.4) = 1.4$ In this convention, z : a larger value gives the candidate more influence.



A state-space recurrence offers compact sequence memory

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Use $\bar{A} = .5$, $\bar{B} = C = 1$ and initial state zero. Inputs 1, 2 and 0 give successive states 1, 2.5 and 1.25. A fixed-dimensional state summarizes the input history.

$$h_t = \bar{A}h_{t-1} + \bar{B}x_t, \quad y_t = Ch_t.$$

$$h_1 = 0.5(0) + 1 = 1, \quad h_2 = 0.5(1) + 2 = 2.5, \quad h_3 = 0.5(2.5) + 0 = 1.25.$$

A fixed-dimensional state summarizes history. Fixed linear dynamics give past inputs structured weights; selective dynamics on the next page also depend on the current input.



Selective state updates can depend on the current input

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A fixed transition weights the past through the same dynamics at every step. Selective models let parts of the update depend on the current input.

$$h_t = \bar{A}_t h_{t-1} + \bar{B}_t x_t, \quad y_t = C_t h_t.$$

Information path	History representation	Choice to examine
Fixed linear state	A fixed-dimensional summary	Dynamics do not change with token content.
Selective state	A fixed-dimensional summary	Some update quantities depend on the input.
Attention	Retrieval from permitted positions	The current query chooses matching content.

Exact parameterization depends on the architecture. These paths do not guarantee the same accuracy or resource cost.

Information access and gradient history are separate choices

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Choose the permitted input before choosing a recurrent architecture.

Setting	Available information	Consequence
Online next-token prediction	Current prefix only	A forward recurrence can use the permitted history.
Whole-sentence representation	The complete given sentence	A bidirectional recurrence may use both directions.
Truncated BPTT	State may continue across chunks	Gradient history stops at a chosen boundary.

Carrying state and resetting gradient history are different operations. Future target information remains excluded from an online prediction.

A query chooses what to retrieve; values supply the content

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A library search separates three roles

Role	Concrete meaning
Query	Request: beginner Python
Key	Book title, topics and level used to match the request
Value	The passages or content returned from a matching book

A strong match decides how much content to retrieve. It is not itself the returned content.

Attention learns these representations

Tokens are rows of $X \in \mathbb{R}^{T \times d_{\text{model}}}$.

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V.$$

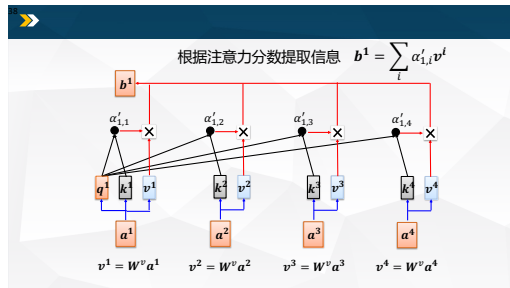
Compare each query with allowed keys, normalize the scores, then take a weighted sum of values.

The library explains the roles. Learned attention need not behave as literal keyword search. The next numeric example shows all three projections.

Read one query path in the original attention figure

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Weights for one query sum to one across candidate positions. Another query generally receives a different weight row.



Scaled dot-product attention has checkable shapes

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Key idea

Every output row is a weighted sum of allowed value rows.

Quantity	Shape	Row and column meanings
Q, K	$T \times d_k$	Position; matching feature
V	$T \times d_v$	Position; retrieved content
$S = QK^T / \sqrt{d_k}$	$T \times T$	Query position; candidate position
$A = \text{softmax}_{\text{row}}(S + M)$	$T \times T$	Row-normalized matching weights
$O = AV$	$T \times d_v$	Retrieved representation at each position

Mask $M_{ij} = 0$ permits access; $M_{ij} = -\infty$ marks blocked access; add it before softmax.



The Q, K and V values come from displayed projections

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

These matrices are supplied to explain the computation. In a trained model their entries are learned.

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad W_Q = W_K = I, \quad W_V = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}.$$

$$Q = XW_Q = X, \quad K = XW_K = X, \quad V = XW_V = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}.$$



Attention example: compute scores and row-wise softmax

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Scores tell us how much content to retrieve. They are not yet the returned content.

$$q = (1, 0) \quad k_1 = (1, 0)$$

$$k_2 = (0, 1) \quad d_k = 2$$

$$\begin{aligned} s_1 &= 1/\sqrt{2} \approx 0.7071, & s_2 &= 0. \\ a_1 &= \frac{e^{0.7071}}{e^{0.7071} + e^0} \approx 0.6698, & a_2 &\approx 0.3302. \end{aligned}$$



Attention example: retrieve the weighted values

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The output combines content from both positions. A larger key score need not imply that every output coordinate is larger.

$$\mathbf{a} = (0.6698, 0.3302) \quad \mathbf{v}_1 = (2, 0)$$

$$\mathbf{v}_2 = (0, 4)$$

$$\mathbf{o} = \mathbf{a}_1 \mathbf{v}_1 + \mathbf{a}_2 \mathbf{v}_2 \approx 0.6698(2, 0) + 0.3302(0, 4) = (1.3396, 1.3208).$$

Item	Role	Numeric result
Keys	Determine relevance	Score 0.7071, 0
Weights	Mix candidates	0.6698, 0.3302
Values	Supply content	(2, 0), (0, 4)
Output	Retrieved representation	(1.3396, 1.3208)



The square-root scale prevents dimension-driven score growth

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

If query and key coordinates are approximately independent, centered and unit-variance, then

$$\text{Var}(qk^{\top}) \approx d_k, \quad \text{Var}(qk^{\top} / \sqrt{d_k}) \approx 1.$$



Mask example: renormalize the legal candidates

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For next-token prediction, query position i may use keys $j \leq i$:

$$M_{ij} = \begin{cases} 0, & j \leq i, \\ -\infty, & j > i. \end{cases}$$

	Key 1	Key 2	Key 3
Query 1	allowed	blocked	blocked
Query 2	allowed	allowed	blocked
Query 3	allowed	allowed	allowed



Changing a future token must not change a past query's causal-attention output.

Access	Softmax weights	Output
Both candidates allowed	(0.6698, 0.3302)	(1.3396, 1.3208)
Second candidate blocked	(1, 0)	(2, 0)

The same words can describe different events

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Same words, different events

Sentence	Position 0	Position 1	Position 2
A	dogs	chase	cats
B	cats	chase	dogs

In A, dogs do the chasing. In B, dogs are chased.
An unordered bag of these words loses this difference.

Without position information or a directional mask, self-attention is permutation-equivariant: reorder input rows, and output rows reorder too.

General sinusoidal encoding:

$$PE_{p,2k} = \sin(p/10000^{2k/d}),$$

$$PE_{p,2k+1} = \cos(p/10000^{2k/d}).$$

Give each position a vector

For a two-dimensional illustration,
 $PE_p = (\sin p, \cos p)$, with p in radians.

Position	First value	Second value
0	0	1
1	.8415	.5403
2	.9093	-.4161

Add the position vector to the token vector. A mask separately controls which positions may be read.

Moving one word changes its position-aware representation

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Move the same word

Give “dogs” a supplied toy token vector

$e = (1, 0)$.

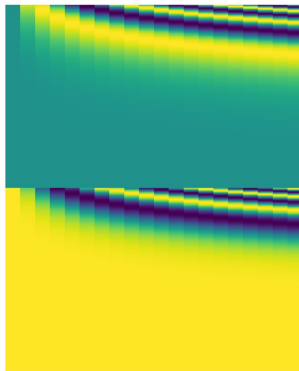
$$e + PE_0 = (1, 0) + (0, 1) = (1, 1).$$

$$\begin{aligned} e + PE_2 &= (1, 0) + (.9093, -.4161) \\ &= (1.9093, -.4161). \end{aligned}$$

The word is unchanged; its position changes the input representation.

The reference heatmap shows position-vector values across positions and channels. Its colors are not attention probabilities. Use the preceding formula for the indexing.

Content and position describe the input. The mask controls permitted access. Training determines how these representations are used.



Multi-head attention combines compatible subspaces

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Each head has its own projections and weight matrix over positions.

$$d_k = d_v = d_{\text{model}}/h$$

$$O_r = \text{Attention}(XW_Q^{(r)}, XW_K^{(r)}, XW_V^{(r)}),$$

$$O = \text{Concat}(O_1, \dots, O_h)W_O.$$

$$d_{\text{model}} = 8 \quad h = 2$$

Check. Concatenation is not averaging. Its output width must match the output projection and residual path.

$d_{\text{model}} = 8, h = 2$ gives width 4 per head. Concatenate the two 4-dimensional outputs, then apply the output projection.



A Transformer block connects retrieval, processing and stable paths

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Cross-token retrieval and within-token feature transformation alternate in a Transformer.

$$\text{FFN}(x) = \phi(xW_1 + b_1)W_2 + b_2.$$

The original Transformer used a post-norm arrangement. Pre-norm and post-norm are different designs; keep one convention consistent when reading a diagram or implementing a block.

$$\begin{aligned}U &= X + \text{MHA}(\text{LN}(X)), \\Y &= U + \text{FFN}(\text{LN}(U)).\end{aligned}$$

Trace one block using shapes and information access

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

At query 2, score columns 3–6 are blocked in every head. Increasing layer count does not remove that boundary.

$$d_{\text{model}} = 8$$

Operation	Shape	What changes
Input plus positions	6×8	Content and order encoded
Each head's score matrix	6×6	Pairwise legal relevance
Each head's value mixture	6×4	Retrieved content
Concatenate and project	6×8	Combined head features
Residual, norm, FFN	6×8	Refined representation, same outer shape



A paraphrase example separates source access from target access

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

An English paraphrase task

Given source: “cats chase dogs”. Desired target: “dogs are chased by cats”. Suppose the generated target prefix is only “dogs are”.

Structure	What this example permits
Encoder self-attention	Read the complete given source: cats, chase, dogs.
Causal decoder self-attention	Read the current target prefix: dogs, are. Future target inputs remain masked.
Cross-attention	A decoder query retrieves encoder-side keys and values from the complete source.

The source is already available. The next target word “chased” is still to be predicted. Full source access and future-target leakage are different information paths.

Here “decoder” names a sequence-model role. It need not be the decoder of a VAE.

Parallel teacher forcing and sequential generation are compatible

Inputs are shifted relative to targets. A causal mask lets all position losses be computed together without revealing their own future targets. During generation, append one predicted token before continuing.

Position	1	2	3
Input	$\langle BOS \rangle$	I	like
Target	I	like	cats
Allowed prefix	BOS	BOS, I	BOS, I, like



The targets are known during training, but the mask blocks future inputs. At generation time, later tokens must be produced sequentially.

Sequence length changes attention cost even with caching

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Pairwise scores

One full-attention head compares each query with every key. Doubling the sequence length quadruples the score count.

Efficient implementations can save storage without removing the pairwise computation.

$$T = 1000 : T^2 = 1,000,000.$$

$$T = 2000 : T^2 = 4,000,000.$$

What KV caching saves

Reuse past key and value projections during generation. The cache still occupies memory, and a new query still matches the preceding keys.

Good reconstructions do not determine every random latent code

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Reconstruct two observed pictures

An ordinary autoencoder uses

$$\mathbf{z} = \mathbf{f}_{\phi}(\mathbf{x}), \quad \hat{\mathbf{x}} = \mathbf{g}_{\theta}(\mathbf{z}),$$

$$L_{\text{AE}} = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \quad (\text{one possible loss}).$$

Suppose picture A is encoded at $\mathbf{z} = -2$ and picture B at $\mathbf{z} = 2$. Both reconstruct well.

Does this guarantee a useful picture at $\mathbf{z} = 0$?

Check where training supplied evidence

Code	What the two reconstructions check
-2	Output near the reconstruction of A
+2	Output near the reconstruction of B
0	No guarantee from those two checks alone

A smooth-looking interpolation is possible, but does not follow from two low reconstruction losses alone.

Random generation needs a rule for which latent values to sample. Next, the VAE specifies a latent prior and a decoder distribution.

One handwritten digit separates encoding, reconstruction and generation

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For a supplied handwritten “2”, suppose the encoder outputs $\mu = 1$ and $\sigma^2 = .25$. These are illustrative encoder values.

$$p(\mathbf{z}) = \mathcal{N}(0, I), \quad q_{\phi}(\mathbf{z} \mid x) = \mathcal{N}(\mu_{\phi}(x), \text{diag } \sigma_{\phi}^2(x)).$$

$$p_{\theta}(x) = \int p_{\theta}(x \mid \mathbf{z}) p(\mathbf{z}) d\mathbf{z}.$$

Use	Start with	Next operation
Encode the observed “2”	The real image x	Obtain $q(\mathbf{z} \mid x) = \mathcal{N}(1, .25)$.
Reconstruct that image	A sample from this $q(\mathbf{z} \mid x)$	Decode to the pixel distribution $p_{\theta}(x \mid \mathbf{z})$.
Generate a new image	No observed image; sample $\mathbf{z} \sim \mathcal{N}(0, 1)$	Decode the sampled $\mathbf{z}$. The digit need not be “2”.

The encoder describes an input-dependent distribution. Pure generation draws from the prior. Next compare these distributions using KL, then calculate a latent sample.

KL compares two probability distributions

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A discrete calculation first

Let $q = (.8, .2)$ and $p = (.5, .5)$ over the same two outcomes. Use natural logarithms.

$$D_{\text{KL}}(q||p) = \sum_z q(z) \log \frac{q(z)}{p(z)}$$

$$= .8 \log(1.6) + .2 \log(.4) \approx .193.$$

If $q = p$, every ratio is 1, so the result is zero.

In the VAE

$q_\phi(z | x)$ is the encoder distribution for one observed input. The prior is $p(z)$.

The penalty compares the whole encoder distribution with the prior. It does not compare one sampled z with zero.

KL is nonnegative and generally asymmetric. For continuous latent values, replace the sum by an integral. Next use the Gaussian formula and calculate one sample.

VAE: calculate KL and a reparameterized sample

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The VAE balances reconstruction likelihood with a KL penalty. Calculate the Gaussian penalty before deriving the training bound.

$$D_{\text{KL}} = \frac{1}{2} \sum_j (\mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1) \quad \text{for the diagonal Gaussian encoder and standard normal prior .}$$

Check. A pixel squared loss corresponds to particular likelihood assumptions; it is not the only reconstruction choice.

Randomness comes from ϵ , while $\mathbf{z}$ is a differentiable transformation of encoder outputs.

$$\mu = 1 \quad \sigma^2 = 0.25$$

$$\sigma = 0.5$$

$$D_{\text{KL}} = \frac{1}{2} [1 + 0.25 - \log(0.25) - 1] \approx 0.818.$$

$$\epsilon \sim \mathcal{N}(0, 1), \quad \mathbf{z} = \mu + \sigma \epsilon.$$

$$\epsilon = 0.4 \quad \mathbf{z} = 1 + 0.5(0.4) = 1.2$$

If otherwise $\mu = 0, \sigma^2 = 1$, the encoder distribution equals the standard-normal prior and KL is zero.



A computable lower bound on log likelihood

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Key idea

The prior $p(z)$ describes latent values before seeing x . The encoder $q_\phi(z | x)$ approximates an input-specific latent posterior. The decoder $p_\theta(x | z)$ scores reconstruction.

$$\begin{aligned}
 D_{KL}(q||p_\theta(z | x)) &= \mathbb{E}_q[\log q - \log p_\theta(z | x)] \\
 &= \mathbb{E}_q[\log q - \log p(z)] \\
 &\quad - \mathbb{E}_q \log p_\theta(x | z) + \log p_\theta(x) \\
 &= D_{KL}(q||p(z)) - \mathbb{E}_q \log p_\theta(x | z) + \log p_\theta(x). \\
 \log p_\theta(x) &= \underbrace{\mathbb{E}_q \log p_\theta(x | z) - D_{KL}(q||p(z))}_{\text{ELBO}} + \underbrace{D_{KL}(q||p_\theta(z | x))}_{\geq 0}.
 \end{aligned}$$



Hence the ELBO is a lower bound on log evidence. Here q always denotes the encoder $q_\phi(z | x)$.

Numerical reading: if $\mathbb{E}_q \log p_\theta(x | z) = -2$ and $D_{KL}(q||p(z)) = .818$, then $\text{ELBO} = -2.818$. The true log evidence is at least this value.

GAN quality needs diversity, not only convincing examples

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Discriminator: separate real and generated

$$\max_D E_{x \sim p_{\text{data}}} \log D(x) + E_z \log[1 - D(G(z))].$$

For one real and one generated example, if $D(x) = .8$ and $D(G(z)) = .3$, its negative objective is

$$L_D = -\log(.8) - \log(.7) \approx .580.$$

Alternate which parameters change. Generator training passes gradients through the discriminator. New-sample generation uses G .

Generator: change the generated example

The common non-saturating loss is

$$L_G = -E_z \log D(G(z)).$$

At generated score .3, $L_G \approx 1.204$. Raising that score to .6 lowers it to .511.

Quality has more than one requirement

Convincing individual samples can still repeat only a few modes. Examine diversity as well as appearance.

Diffusion creates a known noising training task

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The model receives both x_t and the time/noise level. Separate stages need not mean separate networks.

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I).$$

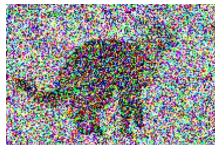
Here $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ is cumulative signal retention. The squared signal and noise coefficients sum to 1. The sampled ϵ supplies the training target.



Forward noising gradually destroys image structure

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Forward noising gradually obscures the image. We sample the added noise ourselves, so it supplies a known training target. Generation starts from noise and applies the learned reverse process.

Clean**Noise 1****Noise 2****Noise 3****Noise 4**

Forward: add sampled noise →

Generation: learn a reverse sampling process ←

A noisy value, a noise target and a clean-value estimate

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Let $x_0 = 0.8$, $\bar{\alpha}_t = 0.64$ and the sampled noise be $\epsilon = -0.5$.

$$x_0 = 0.8 \quad \bar{\alpha}_t = 0.64$$

$$\epsilon = -0.5$$

$$x_t = 0.8(0.8) + 0.6(-0.5) = 0.34.$$

$$\epsilon_\theta(x_t, t) = -0.4$$

$$L_{\text{noise}} = (\epsilon - \epsilon_\theta(x_t, t))^2 = (-0.5 + 0.4)^2 = 0.01.$$

$$\hat{x}_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta}{\sqrt{\bar{\alpha}_t}} = \frac{0.34 - 0.6(-0.4)}{0.8} = 0.725$$

At a chosen noise level, use the denoiser and the sampler's update rule.



Compare generative families by training signal and sampling

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Assess broad quality, diversity, task suitability and memorization with an appropriate evaluation design. A few selected attractive samples do not establish distribution learning.

Model	Training signal	Sampling procedure
Autoregressive	Observed next token	Sample sequentially from the prefix
VAE	Reconstruction likelihood and latent KL	Sample a latent from the prior; decode it
GAN	Discriminator feedback to the generator	Sample noise; apply the generator
Diffusion	Known noise added during training	Start from noise; denoise over multiple steps

Condition to check

Attractive samples do not establish distribution coverage. Examine diversity, task performance, memorization and sampling cost.

Which information and training target does the model use?

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Question

Before choosing RNN, attention, VAE or diffusion, state what the model may observe and what supplies its training target.

Evidence and reasoning

An RNN compresses preceding inputs into state; causal attention retrieves only permitted positions. A VAE learns reconstruction with latent regularization. Diffusion predicts a known noising target and samples through a learned reverse process.

Check. Compare evidence for the intended task. Attractive samples or large context capacity alone do not establish reliability.