Multimodal dialogue models

Artem Chervyakov

Intro

Outline:

1. Recap on attention and Transformers

2. How to encode Images: ViT

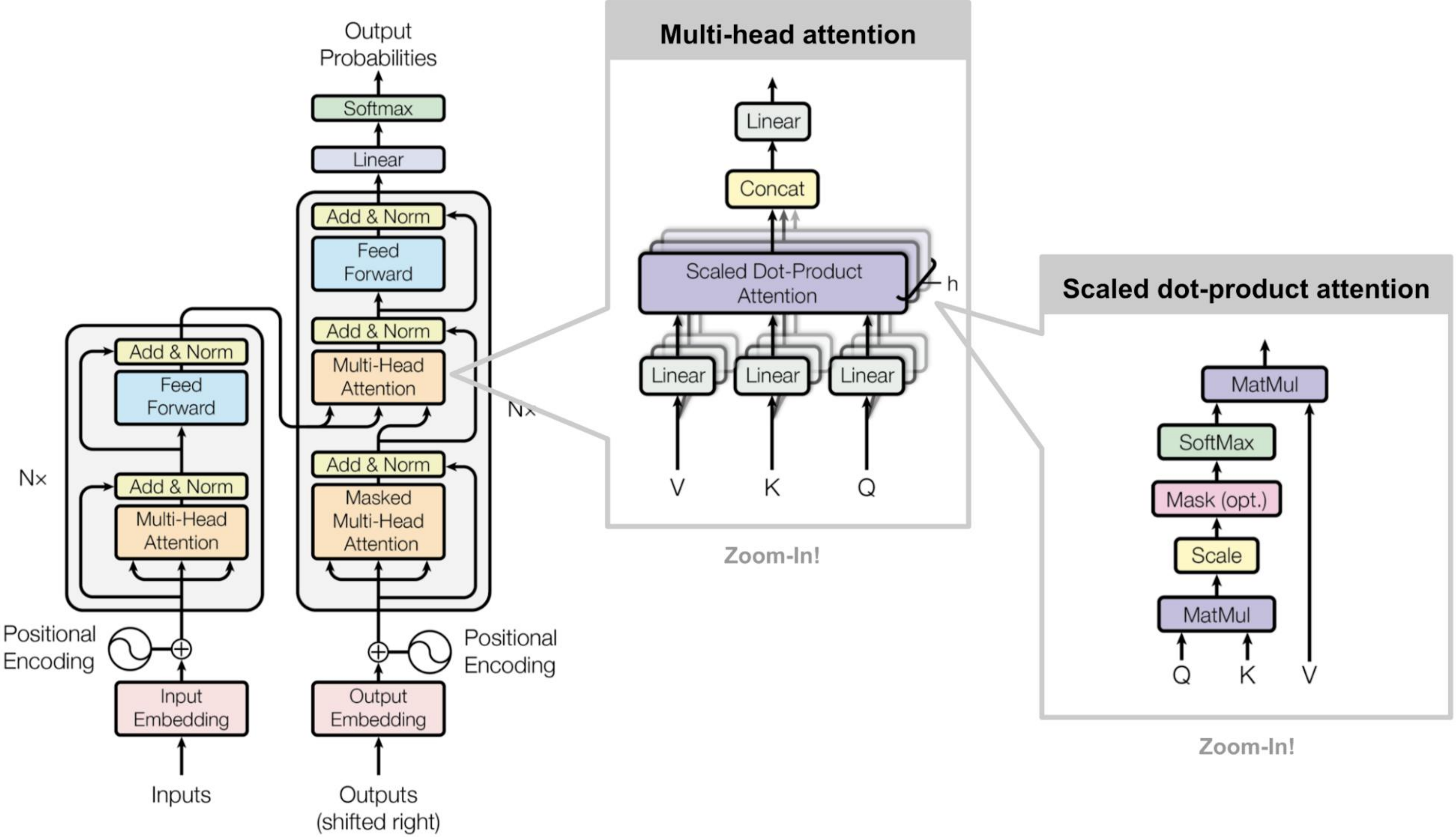
3. Image-to-text models: CLIP, Qwen VL

4. Text-to-image models: Kandinsky

5. What about other modalities?

Recap on attention and Transformers

Recap on attention and Transformers

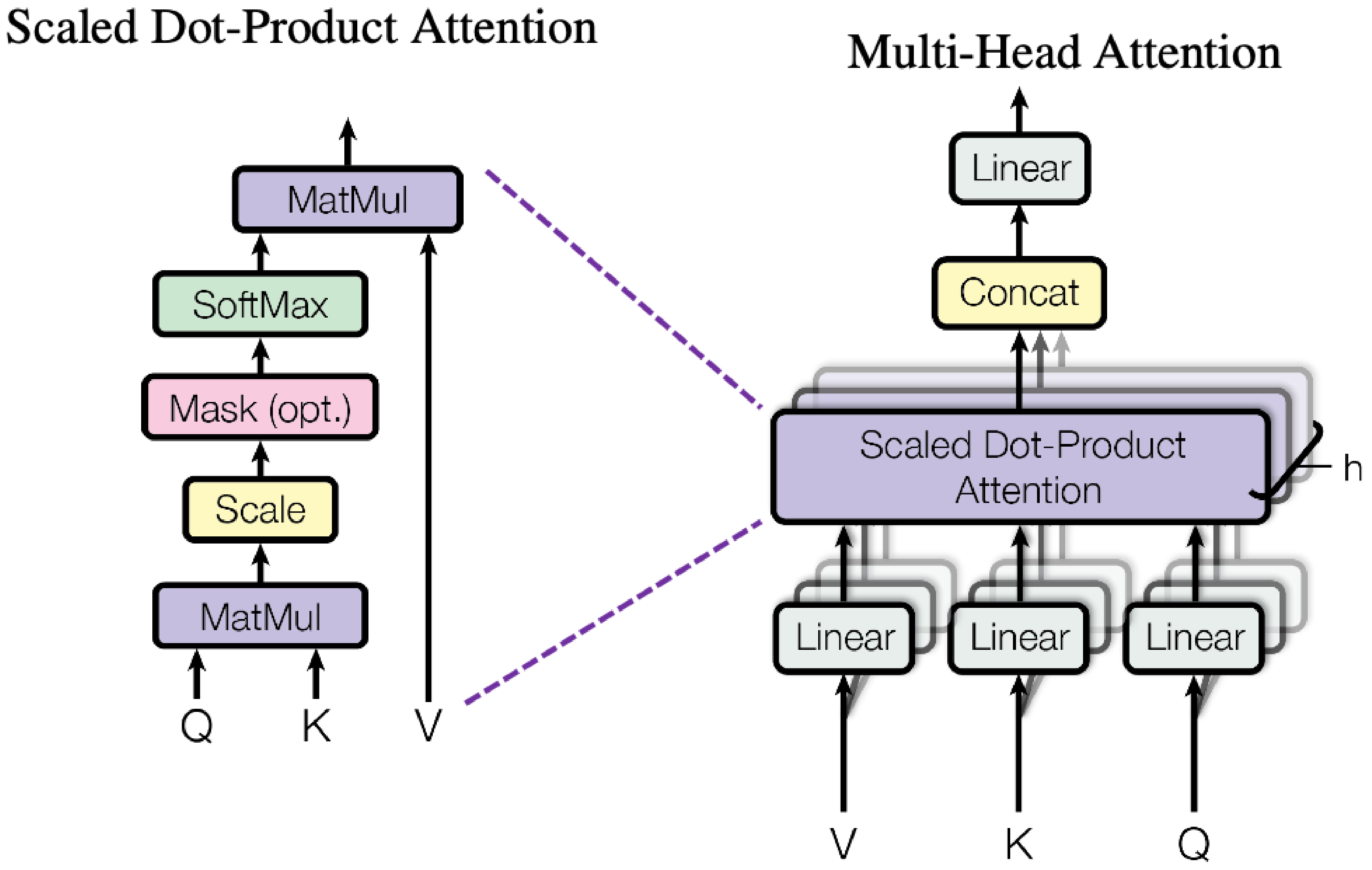


https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Recap on attention and Transformers

Each token has three representations:
- as query
- as key
- as value

$$attn = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

How to encode Images: ViT

How to encode Images: ViT

We need to somehow encode the image: represent its contents as a vector. How?

How to encode Images: ViT

We need to somehow encode the image: represent its contents as a vector. How?

Can we use Feed-Forward network? Yes, but it is extremely prone to overfitting and not invariant to image shift.

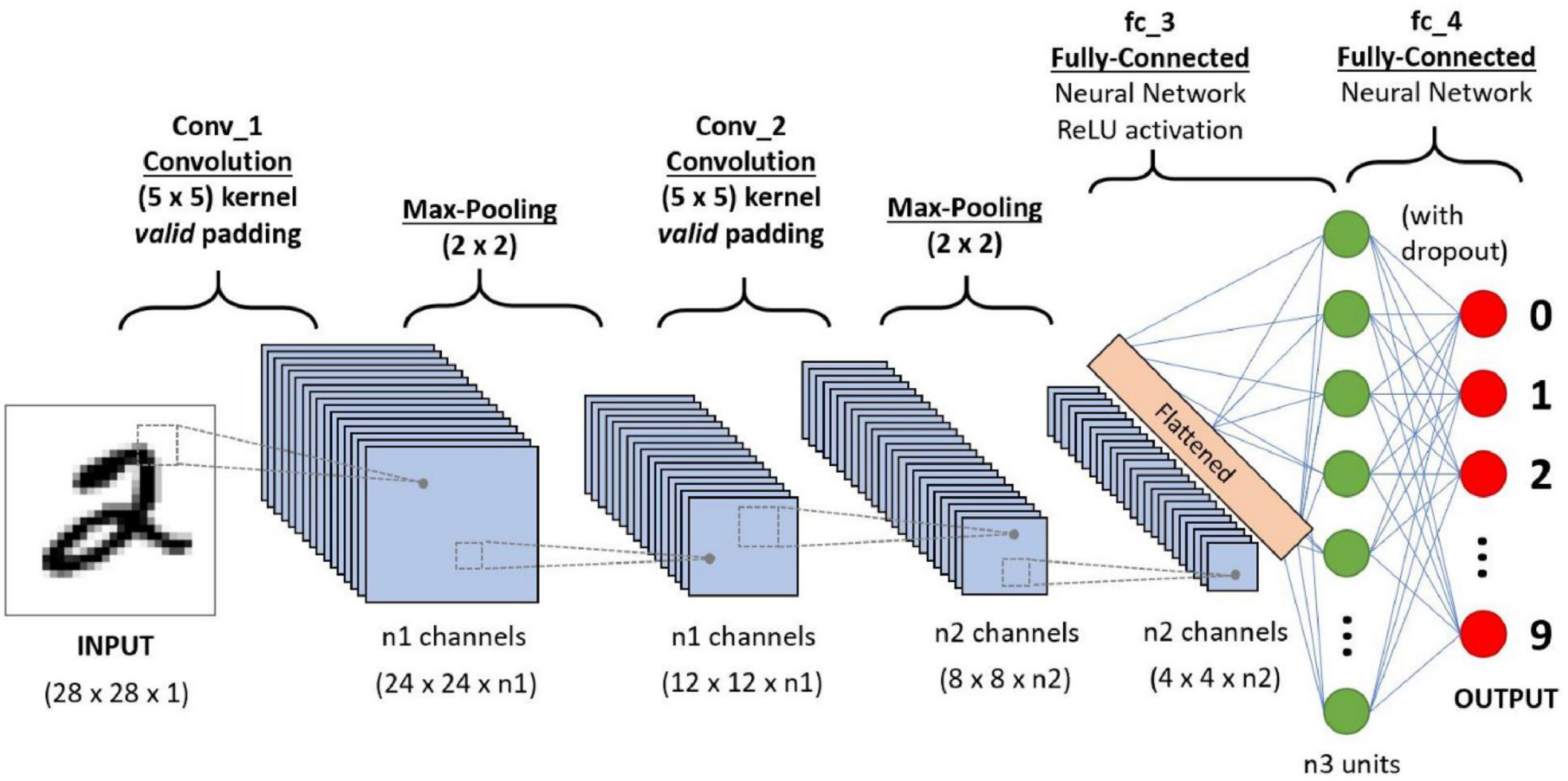
How to encode Images: ViT

We need to somehow encode the image: represent its contents as a vector. How?

Can we use Feed-Forward network? Yes, but it is extremely prone to overfitting and not invariant to image shift.

Then we can use Convolutional Networks! Yes, but only for feature extraction.

How to encode Images: ViT



https://pubmed.ncbi.nlm.nih.gov/35062530/

How to encode Images: ViT

Well, we can make an image to become a vector of features.
Can we now use modern NLP architectures to handle it?

How to encode Images: ViT

Well, we can make an image to become a vector of features.
Can we now use modern NLP architectures to handle it?

No, we cannot. All NLP architectures assume the input data is a sequence – ordered chain of elements that are interconnected.

How to encode Images: ViT

Well, we can make an image to become a vector of features.
Can we now use modern NLP architectures to handle it?

No, we cannot. All NLP architectures assume the input data is a sequence – ordered chain of elements that are interconnected.

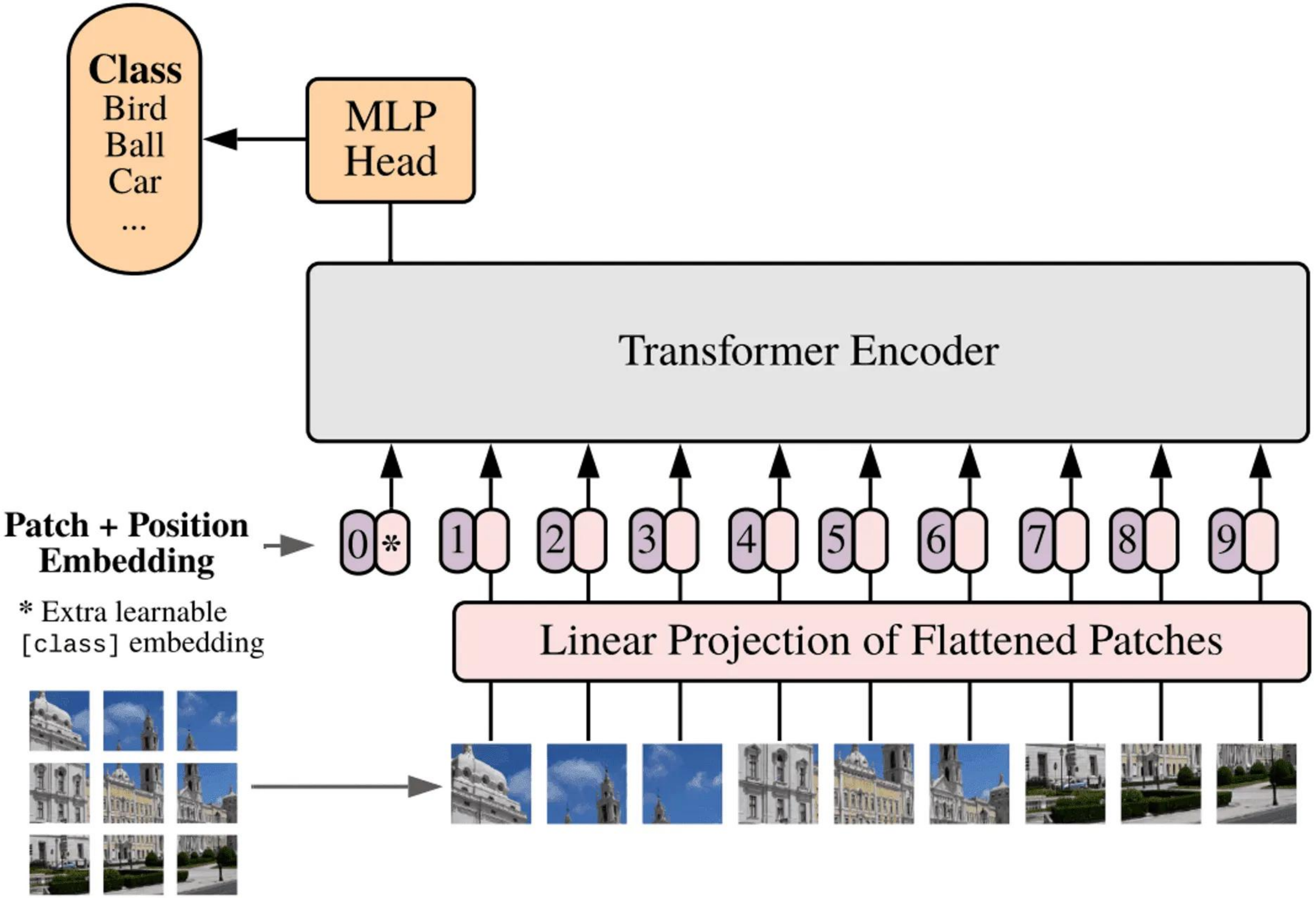
So, we need to somehow make one image become a sequence of elements to handle them with Transformers.
And here ViT emerges...

How to encode Images: ViT

Split image into patches.

Consider each patch to be some token.

Encode each patch and deal with sequence of tokens.



https://arxiv.org/pdf/2010.11929

How to encode Images: ViT

1. Split image $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$ into N patches of $P \times P$ size (x_p).

2. Flatten each patch: $x_p \in \mathbb{R}^{P^2 C}$

3. Apply a linear projection: $z_p = x_p E$, where $E \in \mathbb{R}^{(P^2 C) \times D}$

4. Prepend a learnable [CLS] token: $Z = [z_{cls}; z_1; z_2; \dots; z_N]$

5. Assign a learnable positional embedding: $E_{pos} \in \mathbb{R}^{(N+1) \times D}$

$$Z = Z + E_{pos}$$

How to encode Images: ViT

6. Project input into queries (Q), keys (K), values (V):

$$Q = ZW^{Q}, K = ZW^{K}, V = ZW^{V}$$

7. Computes attention scores:

$$attn(Q, K, V) = softmax\left(\frac{QK^{T}}{\sqrt{d_k}}\right)V$$

M. After L layers use [CLS] token for classification:

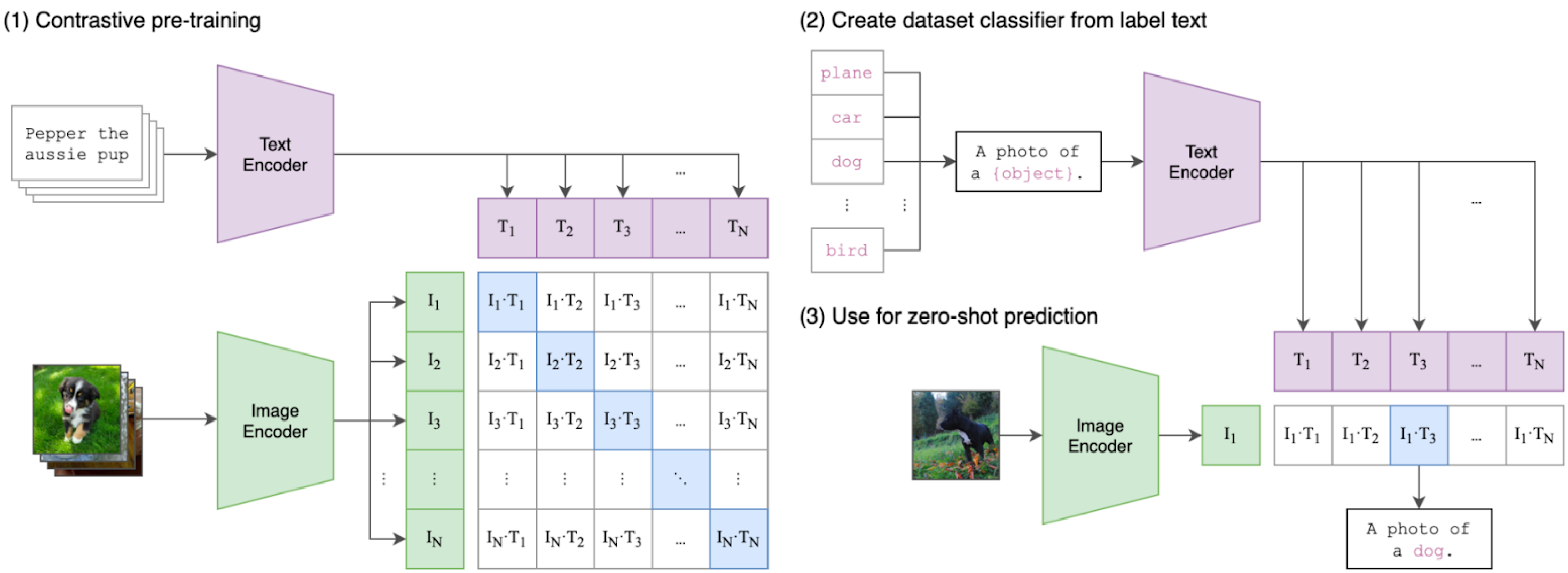
$$\tilde{y} = MLP\left(z_{cls}^{L}\right)$$

M+1. Compute Cross-Entropy Loss:

$$L(\tilde{y}, y) = -\sum_{i=1}^{k} y_i \log(\tilde{y}_i)$$

Image-to-text models: CLIP, Qwen VL

Image-to-text models: CLIP, Qwen VL



https://arxiv.org/pdf/2103.00020

Image-to-text models: CLIP, Qwen VL

1. Extract from image $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$ embedding of [CLS] with ViT:

$$v \in \mathbb{R}^d$$

2. Extract from text embedding of [EOS] with Transformer:

$$t \in \mathbb{R}^d$$

3. Compute similarity matrix: $V = [v_1, \dots, v_N], T = [t_1, \dots, t_N]$

$$S_{i,j} = v_i \cdot t_j \text{ (cosine similarity)}$$

4. Contrastive Loss:

$$\mathcal{L}_{image \to text} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp\left(\frac{S_{i,i}}{\tau}\right)}{\sum_{j=1}^{N} \exp\left(\frac{S_{i,j}}{\tau}\right)}$$

Image-to-text models: CLIP, Qwen VL

4. Contrastive Loss:

$$\mathcal{L}_{text \to image} = -\frac{1}{N}\sum_{j=1}^{N} \log \frac{\exp\left(\frac{S_{j,j}}{\tau}\right)}{\sum_{i=1}^{N} \exp\left(\frac{S_{i,j}}{\tau}\right)}$$

$$\mathcal{L}_{total} = \frac{1}{2}\left(\mathcal{L}_{image \to text} + \mathcal{L}_{text \to image}\right)$$

We want the classifier to predict the same class for image and text from the same pair. Hence, we want their embeddings to be similar (in terms of cosine similarity).

Image-to-text models: CLIP, Qwen VL

How to predict the class?

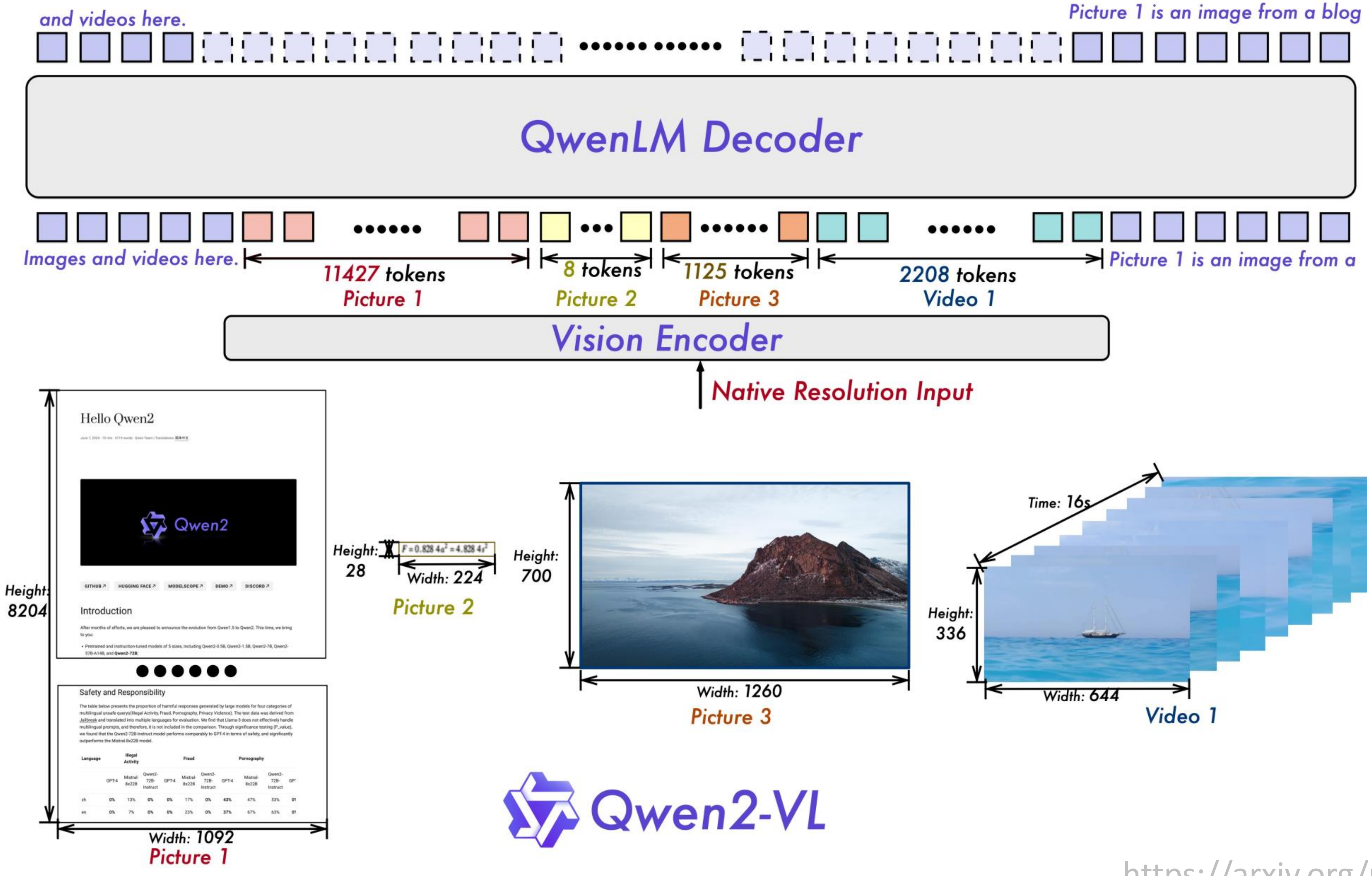
1. Given the image $\mathcal{X}$ we encode it with vector v.

2. Produce the prompt for possible classes: "a photo of a {dog, cat, car, …}" and encode them: $t_1, t_2, \dots, t_k$.

3. Predict class with highest similarity:

$$\tilde{y} = \arg\max_k (v \cdot t_k)$$

Image-to-text models: CLIP, Qwen VL



https://arxiv.org/pdf/2409.12191

Image-to-text models: CLIP, Qwen VL

1. Extract from image $\mathcal{X}$ patches embeddings with ViT:

$$Z_{img} \in \mathbb{R}^{N\times D}\text{, where } N = \text{number of patches}$$

2. Map text tokens (from Qwen tokenizer) to embeddings:

$$Z_{text} \in \mathbb{R}^{L\times D}\text{, where L = sequence length}$$

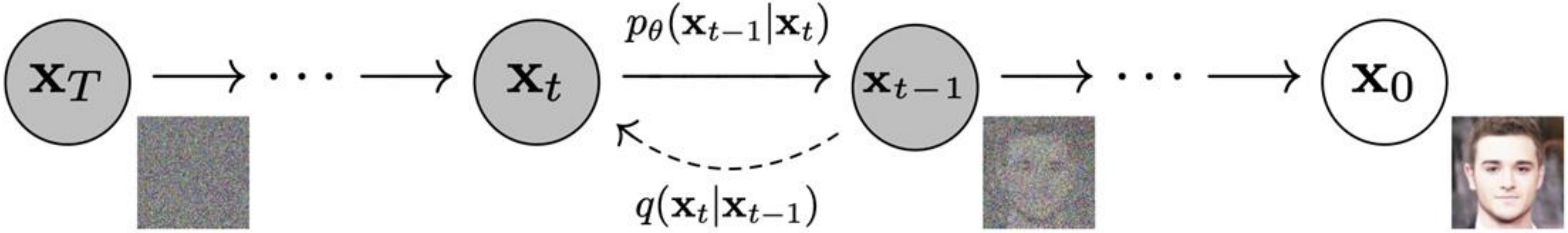
3. Cross-Modal Fusion:

$$Z_{joint} = [Z_{img}, Z_{text}]$$

Now we use the regular attention mechanism, but Queries (Q) are taken only from text tokens. So, text tokens may “look” at the images while generating new text.

Text-to-image models: Kandinsky

Text-to-image models: Kandinsky

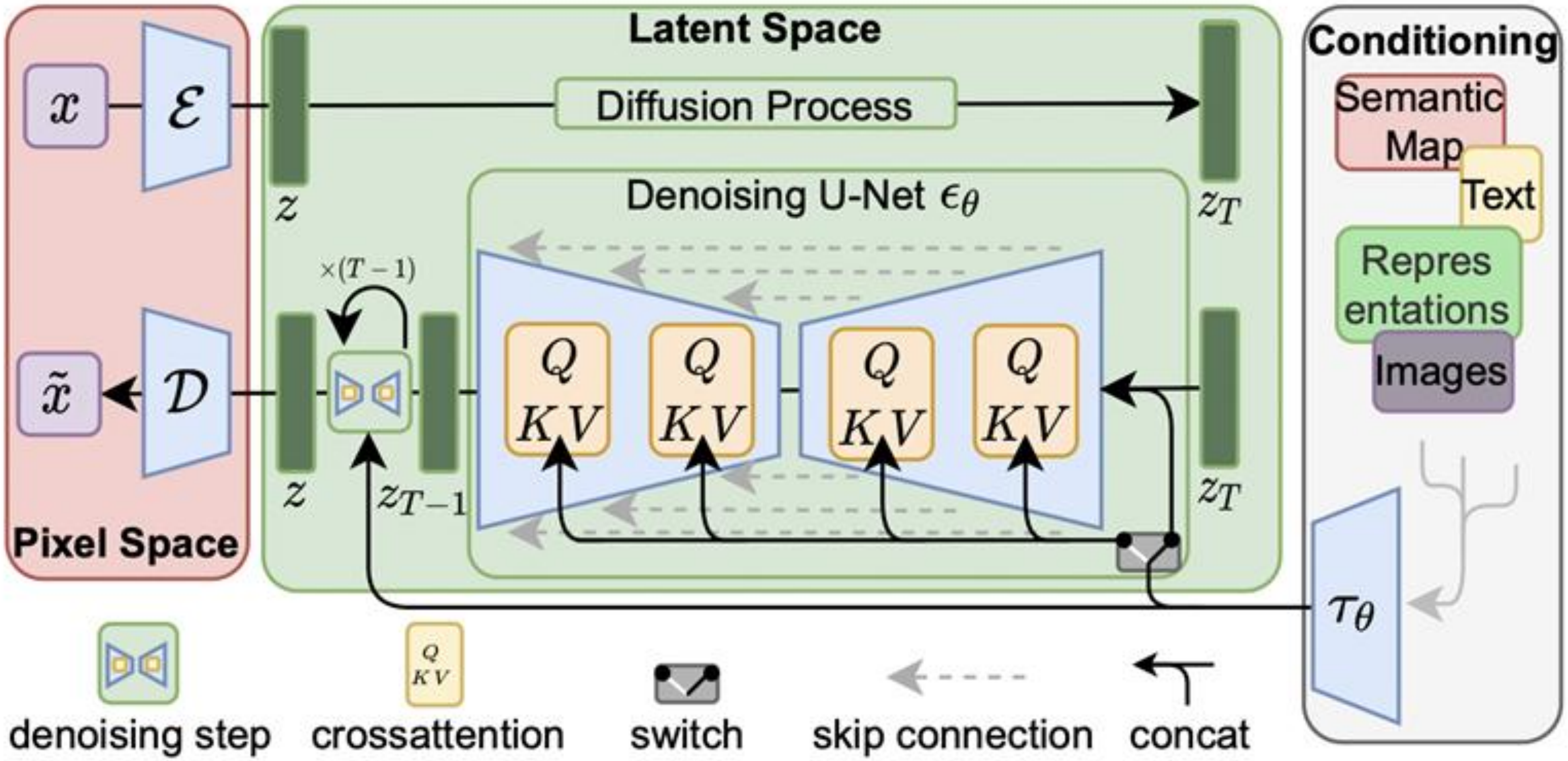


Diffusion is all about making an image step-by-step. On each step we do not require the model to make the final image, but take the image from previous step and make it better (more clear, less noisy).

Starting from the pure white noise image we do N steps and get the final image based on text condition.

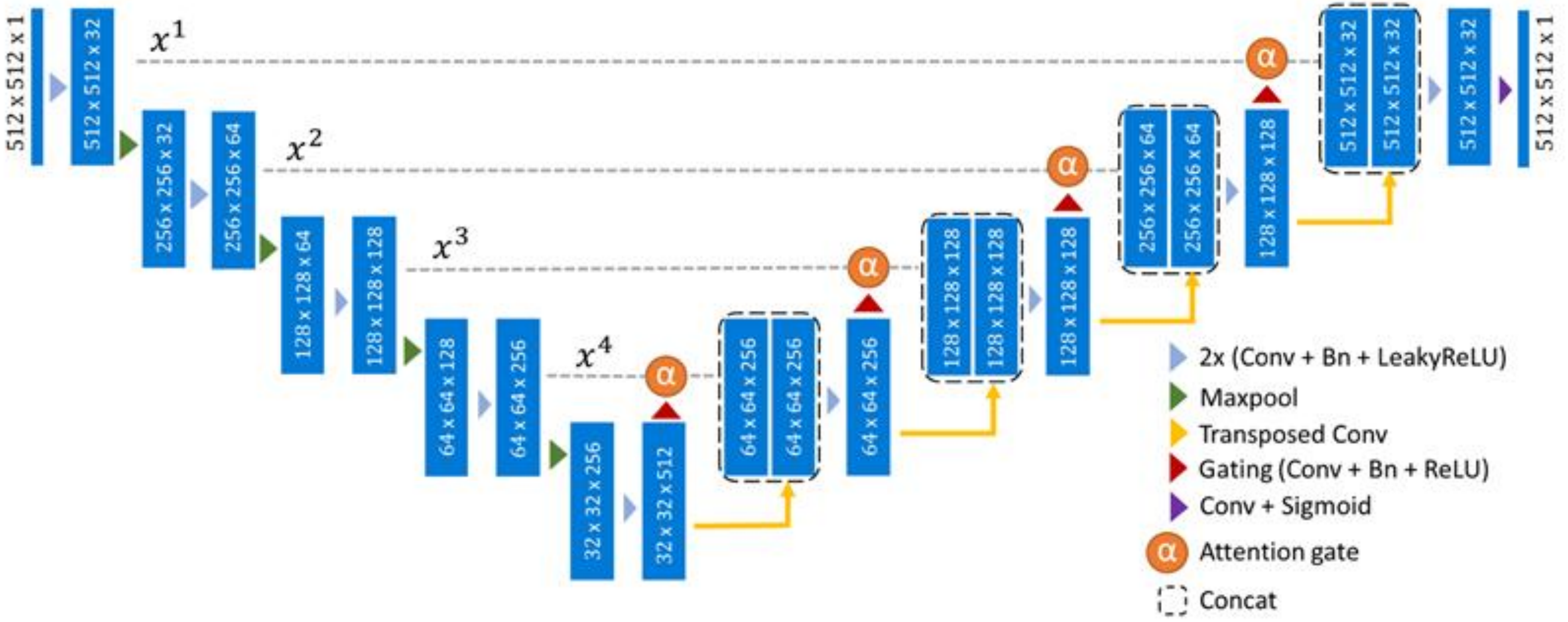
https://arxiv.org/pdf/2006.11239

Text-to-image models: Kandinsky



https://arxiv.org/pdf/2112.10752

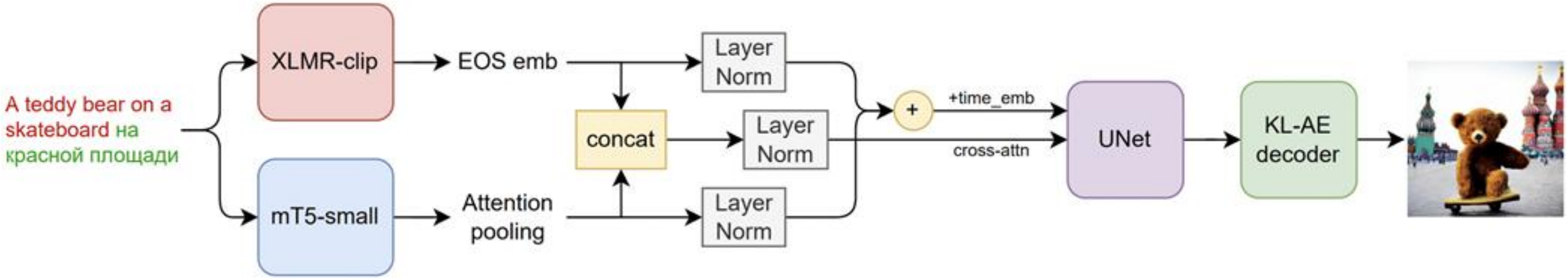
Text-to-image models: Kandinsky



U-net convolves and then up-samples the images (to make one denoising step). Some layers may include Attention. For example, consider layer output as Queries and take text token Keys and Values to “condition” images on text.

https://habr.com/ru/companies/sberbank/articles/701162/

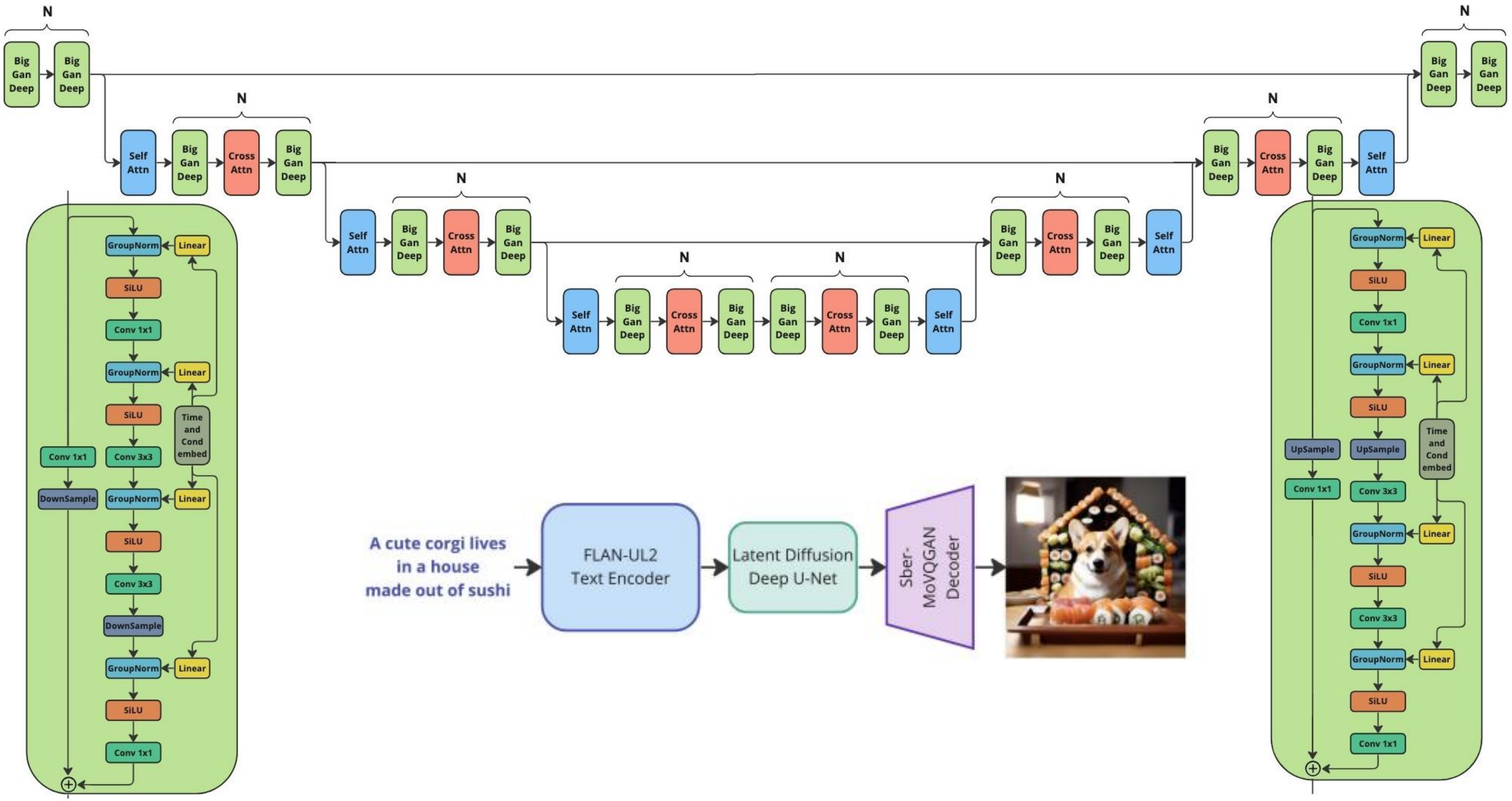
Text-to-image models: Kandinsky



Kandinsky 2.0 encodes texts with CLIP and mT5 models, then uses U-Net to denoise the image and get the intermediate final image representation. The final image is decoded by KL-AE from intermediate representation.

https://habr.com/ru/companies/sberbank/articles/701162/

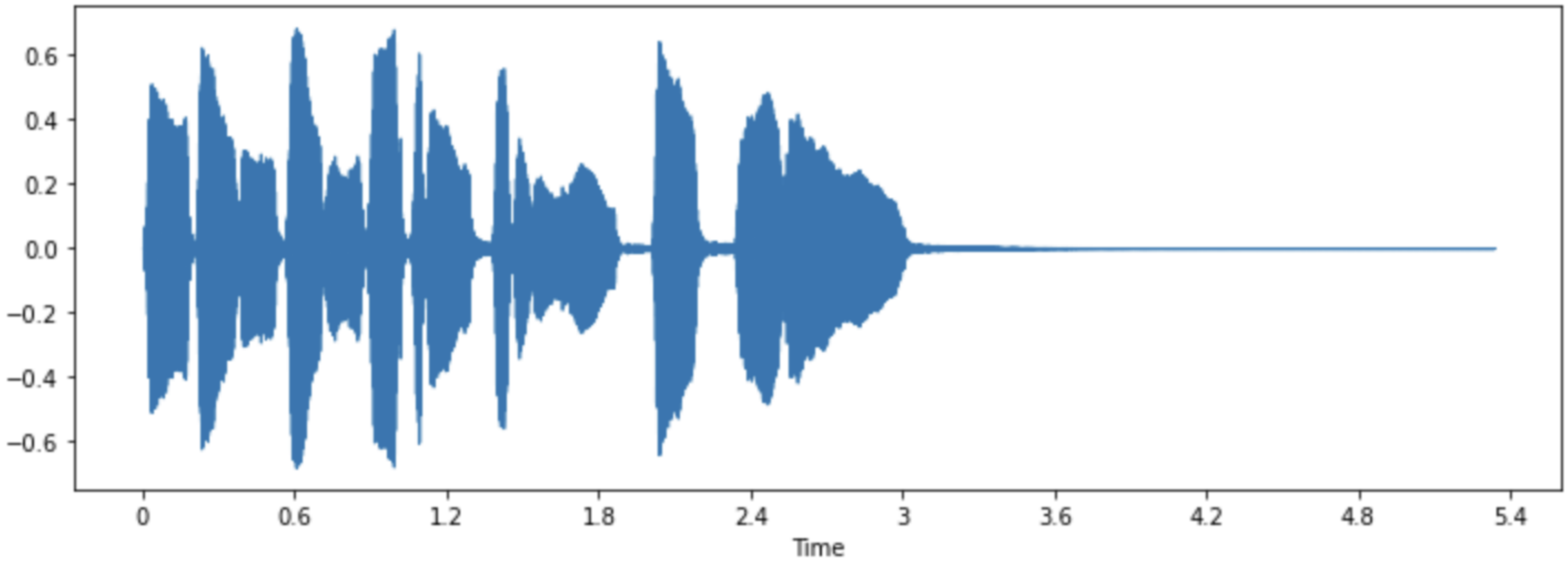
Text-to-image models: Kandinsky



https://arxiv.org/pdf/2312.03511

What about other modalities?

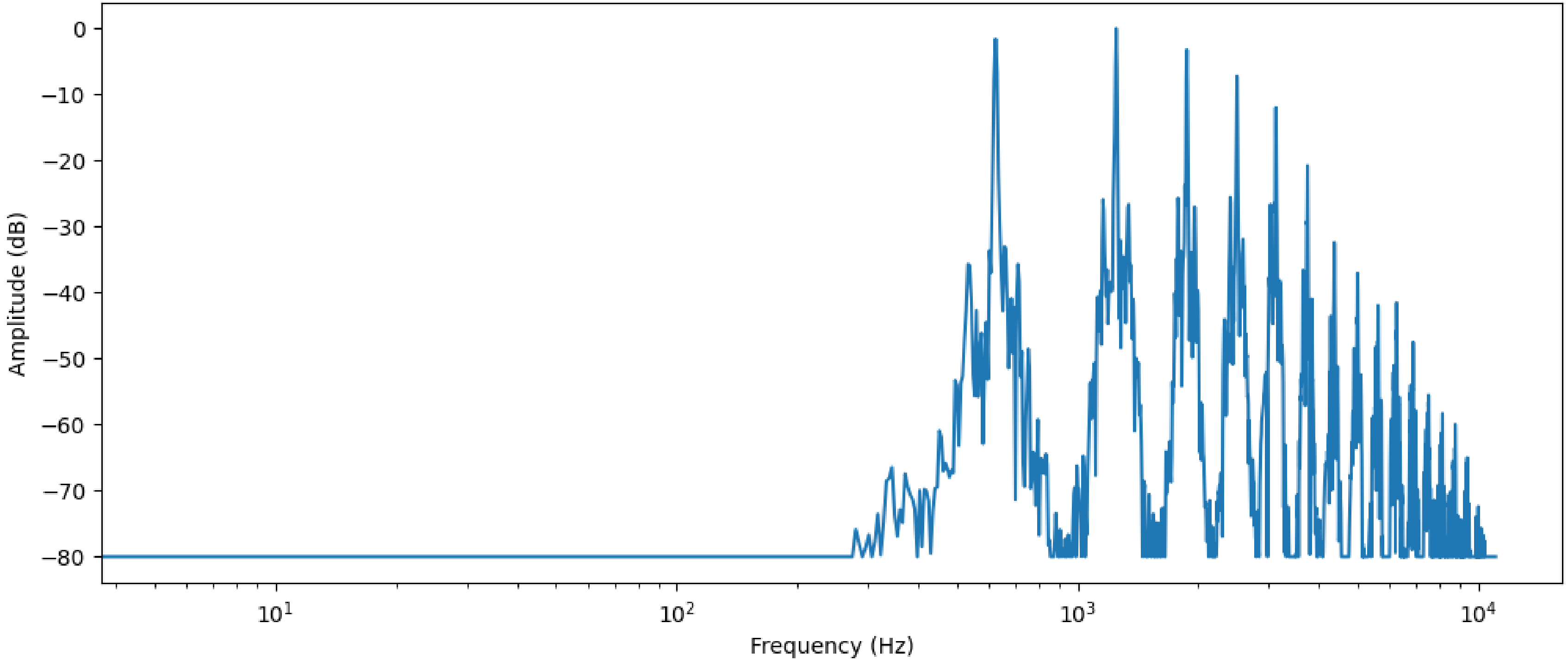
What about other modalities?



Audio is just some waves along the time. We can draw it.

https://huggingface.co/learn/audio-course/ru/chapter1/audio_data

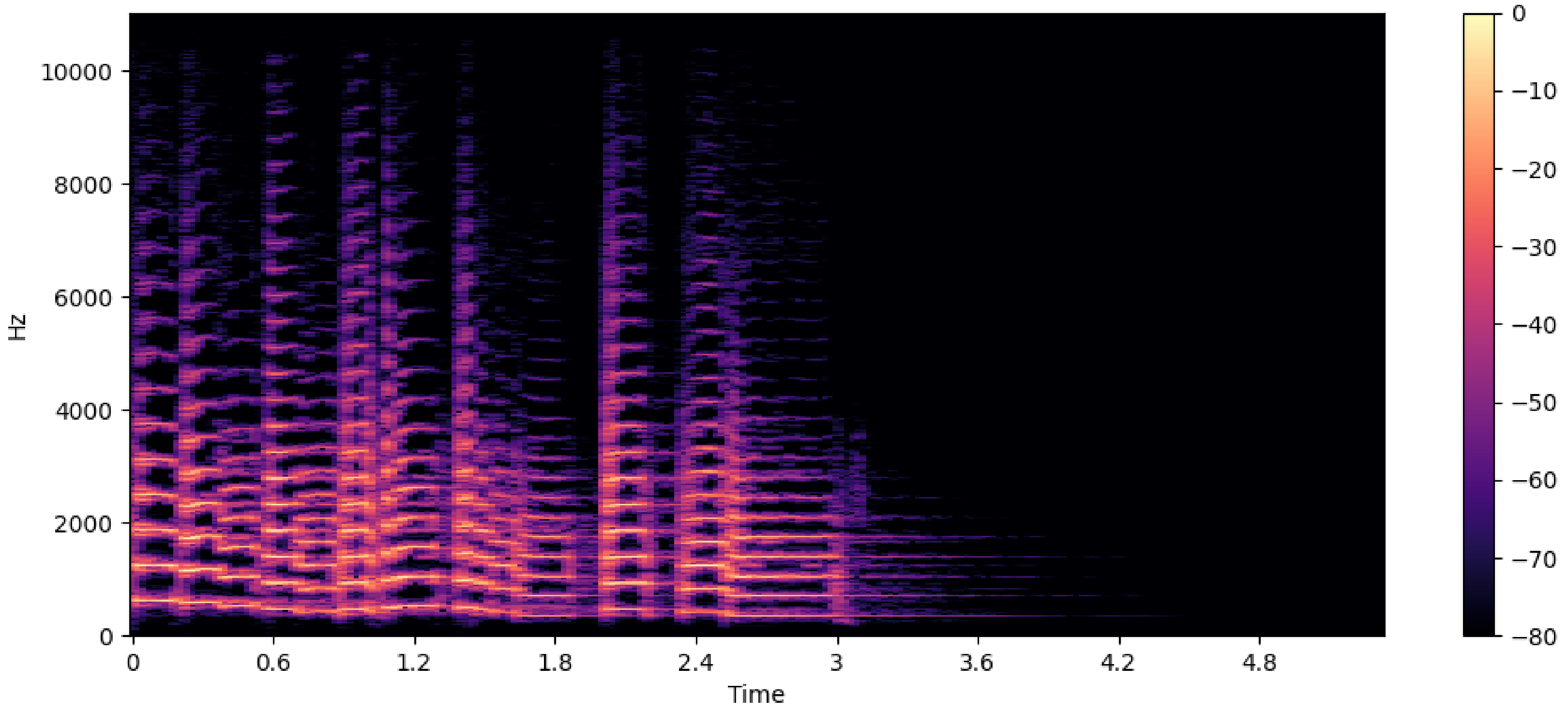
What about other modalities?



We can present it as the frequencies and their strength (intensity).

https://huggingface.co/learn/audio-course/ru/chapter1/audio_data

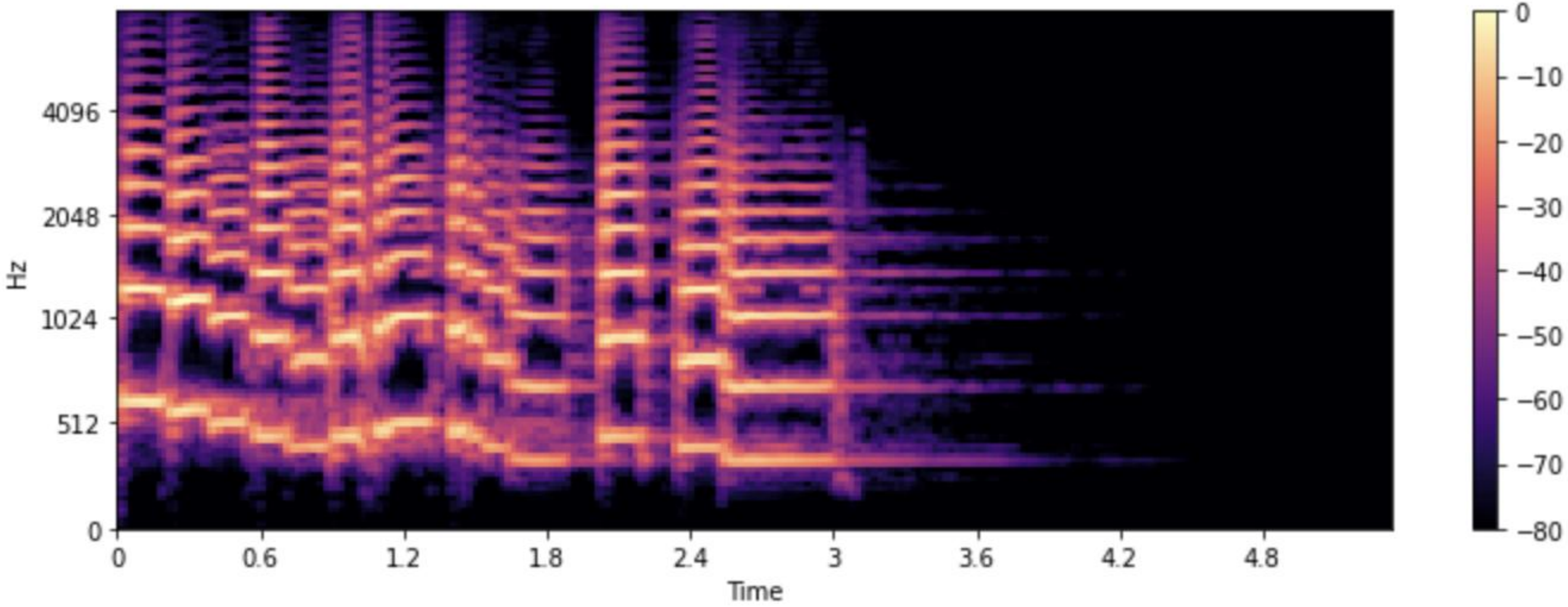
What about other modalities?



For a period of time, we can take specters and combine them into one spectrogram.

https://huggingface.co/learn/audio-course/ru/chapter1/audio_data

What about other modalities?



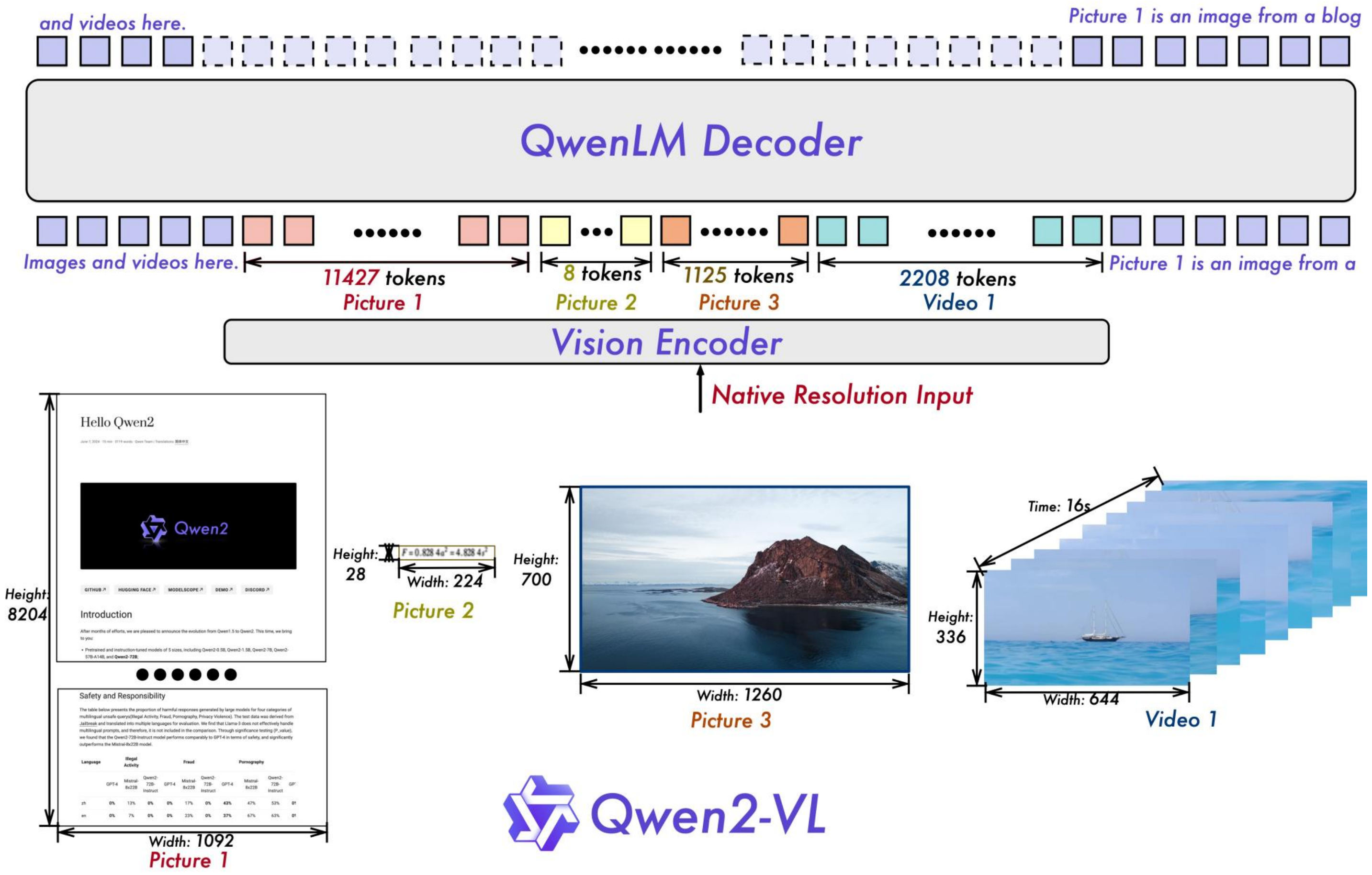
Mel - spectrogram adjusts to the human hearing system.

https://huggingface.co/learn/audio-course/ru/chapter1/audio_data

What about other modalities?

Video is a sequence of images and optionally audio.

Difference – there is the temporal axis to track the sequence of frames.



https://arxiv.org/pdf/2409.12191

Thank you for your attention!

Artem Chervyakov

Middle ML-engineer, SBER

@arorlov