Report paper: Neural Discrete Representation Learning

Giáp Ngọc Hiệu

**Tóm tắt nội dung**

Bài report này là phần tóm tắt lại ý tưởng, nội dung chính, những điều làm được của paper

Neural Discrete Representation Learning

1. **Ý tưởng**

Học các cách biểu diễn hữu ích mà không có sự giám sát vẫn là một thách thức chính trong học máy. Trong bài báo này, tác giả đề xuất một mô hình sinh mạnh mẽ và đơn giản gọi là Vector Quantised-Variational AutoEncoder (VQ-VAE) để học các biểu diễn hữu ích mà không cần sự giám sát. Mô hình này khác biệt với Variational AutoEncoders (VAEs) ở hai điểm chính:

* Mã hóa đầu ra dưới dạng mã rời rạc: Trong VQ-VAE, mạng mã hóa (encoder) tạo ra các mã rời rạc thay vì mã liên tục. Điều này giúp tạo ra các biểu diễn dễ quản lý và giúp giảm bớt một số vấn đề phức tạp trong việc đối phó với dữ liệu liên tục.
* Prior được học chứ không phải là cố định: Trong VQ-VAE, prior distribution của biến ẩn được học từ dữ liệu thay vì được giả định là một phân phối cố định như trong các mô hình truyền thống.

Để học biểu diễn biến ẩn rời rạc, mô hình kết hợp các ý tưởng từ vector quantisation (VQ). Sử dụng phương pháp này giúp mô hình tránh được các vấn đề của "posterior collapse", tức là sự bỏ qua các biến ẩn khi chúng được kết hợp với một mạng decoder tự hồi quy mạnh mẽ, thường quan sát được trong VAE. Bằng cách kết hợp các biểu diễn này với một prior tự hồi quy, mô hình có thể sinh ra các hình ảnh, video và âm thanh chất lượng cao, cũng như thực hiện chuyển đổi loa chất lượng cao và học không giám sát các phương pháp. Điều này cung cấp thêm bằng chứng về sự hữu ích của các biểu diễn đã học được.

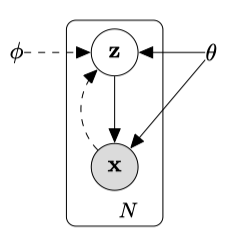
\* "Posterior collapse" là một vấn đề phổ biến trong mô hình Variational AutoEncoder (VAE). Nó xuất hiện khi phân phối posterior, được ước lượng bởi mạng encoder, không còn chứa nhiều thông tin hữu ích về dữ liệu đầu vào và thay vào đó chỉ trở thành một phân phối tương tự nhau, đồng nhất hoặc không đổi trong suốt quá trình huấn luyện. Nguyên nhân chính của posterior collapse là mạng encoder học được một phép ánh xạ tương ứng với một phân phối posterior tại mỗi điểm dữ liệu đầu vào, nhưng không học được một phân phối có biến thiên đủ lớn để phản ánh các đặc trưng khác nhau của dữ liệu. Khi điều này xảy ra, mạng decoder không còn cần sử dụng thông tin từ biến ẩn và chỉ phụ thuộc vào nó để sinh ra dữ liệu mới. Kết quả của posterior collapse là mô hình không thể học được biểu diễn dữ liệu hiệu quả từ các biến ẩn, và thường dẫn đến việc sinh ra dữ liệu mới không chất lượng và không đa dạng.

Những đóng góp chính của tác giả:

* Giới thiệu mô hình VQ-VAE, một mô hình đơn giản sử dụng các biến ẩn rời rạc, không gặp vấn đề "posterior collapse" và không có vấn đề về phương sai. Điều này giúp cải thiện tính linh hoạt và hiệu suất của mô hình.
* Chứng minh rằng một mô hình biến ẩn rời rạc (VQ-VAE) hoạt động cũng tốt như các mô hình liên tục tương đương của nó khi đánh giá bằng log-likelihood. Điều này chứng tỏ tính hiệu quả của việc sử dụng biến ẩn rời rạc trong quá trình huấn luyện.
* Khi kết hợp với một prior mạnh mẽ, các mẫu sinh ra từ mô hình có tính nhất quán và chất lượng cao trên một loạt các ứng dụng như sinh ra tiếng nói và video. Điều này làm tăng tính ứng dụng của mô hình trong các bài toán sinh dữ liệu.
* Cung cấp bằng chứng cho thấy khả năng học ngôn ngữ thông qua tiếng nói trực tiếp, mà không cần bất kỳ sự giám sát nào. Đồng thời, trình bày các ứng dụng của việc chuyển đổi giọng nói không giám sát, là một ứng dụng tiềm năng của mô hình VQ-VAE trong lĩnh vực xử lý tiếng nói.

1. **Một số kiến thức cần thiết**

**2.1 VAE (Variational Auto-Encoder)**



Luồng encoder (được mô tả bằng đường chấm) thể hiện việc nén hình ảnh đầu vào x thành biểu diễn tiềm ẩn z trong không gian đặc trưng. Ký hiệu ϕ đại diện cho các tham số của Encoder.



Luồng decoder (được biểu thị bằng đường liền nét) minh họa quá trình tái cấu trúc, chuyển đổi biểu diễn tiềm ẩn z thành hình ảnh giống với đầu vào x ban đầu. Ở đây, θ biểu thị các tham số của Decoder.



Encoder và Decoder có tính xác suất và phân bố xác suất của chúng thể hiện sự phụ thuộc có điều kiện, vì hai luồng có kết nối thông qua biểu diễn tiềm ẩn z.



Encoder chịu trách nhiệm mô hình hóa phân bố xác suất có điều kiện P(z|x), vì nó lấy hình ảnh đầu vào x và tạo ra biểu diễn tiềm ẩn z. Decoder chịu trách nhiệm mô hình hóa phân bố xác suất có điều kiện P(x|z), vì nó lấy các biến tiềm ẩn z và tái tạo lại một hình ảnh x' gần với hình ảnh đầu vào x.

**Continuous Latent Variables: Hidden Features**

Trong ngữ cảnh của VAE, biến z nắm bắt các đặc trưng tiềm ẩn cơ bản của dữ liệu đầu vào. Mặc dù các khía cạnh riêng lẻ của một hình ảnh, chẳng hạn như màu sắc hoặc hình dạng cụ thể, có thể được nhận thấy trực tiếp, nhưng các đặc trưng tiềm ẩn sẽ tìm hiểu sâu hơn, thể hiện sự trừu tượng ở cấp độ cao hơn được suy ra từ dữ liệu thô. Những đặc điểm trừu tượng này, thường ẩn trong các pixel nhưng nhất quán trên nhiều hình ảnh, nắm bắt các đặc điểm chung trên nhiều đầu vào khác nhau. Encoder chắt lọc hình ảnh đầu vào x thành biểu diễn trừu tượng này, vượt qua các giá trị pixel tức thời để nắm bắt được bản chất của nội dung.

Không gian tiềm ẩn nổi bật vì hai đặc điểm chính:

* Compactness (Tính nhỏ gọn) : Không gian tiềm ẩn của VAE thường có chiều thấp hơn dữ liệu gốc. Sự biểu diễn nhỏ gọn này đảm bảo VAE tập trung vào các tính năng nổi bật và có tính khái quát nhất. Bằng cách giảm kích thước, mô hình trừu tượng hóa khỏi các chi tiết, pixel, tập trung sự chú ý vào bản chất bao quát của nội dung.
* Continuity (Tính liên tục) : Các biến tiềm ẩn z là liên tục, cho phép chuyển tiếp trơn tru và mạch lạc trong không gian tiềm ẩn. Tính liên tục này đảm bảo rằng thậm chí những biến động nhỏ nhất trong các biến ẩn có thể được ánh xạ thành các biến đổi có ý nghĩa trong đầu ra được tạo ra, cho phép VAE biểu diễn một loạt rộng lớn các đặc điểm. Khi decoder tận dụng các biến tiềm ẩn liên tục này, nó sẽ tạo ra các hình ảnh phù hợp với phong cách và nội dung của bản gốc mà không chỉ là bản sao.

Cùng với nhau, những đặc điểm này của không gian tiềm ẩn cho phép VAE tạo ra các mẫu đa dạng và mới lạ trong khi vẫn giữ được các thuộc tính cơ bản của dữ liệu huấn luyện.

Giả sử chúng ta có thể xác định sự phân bố của các biến tiềm ẩn này và lấy mẫu từ không gian tiềm ẩn. Trong trường hợp đó, chúng ta có thể sử dụng decoder để tạo ra các hình ảnh mới phản ánh phong cách và không gian của tập dữ liệu huấn luyện mà không cần hình ảnh đầu vào hoặc encoder.

Tuy nhiên, vẫn tồn tại một thách thức đáng kể: việc xác định chính xác sự phân bố của các biến tiềm ẩn z, không chỉ phức tạp mà còn khó giải quyết.

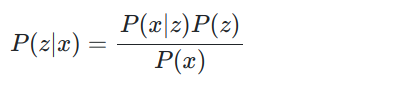
**Prior và posterior distribution**

Trong VAE, cả prior và posterior distribution đều là các thành phần quan trọng của mô hình và chúng phục vụ các mục đích khác nhau:

* Phân Phối Prior: Phân phối prior P(z) đại diện cho niềm tin của chúng ta về các biến ẩn z trước khi quan sát bất kỳ dữ liệu nào. Trong một VAE, phân phối prior thường được chọn là một phân phối đơn giản, như phân phối chuẩn N(0,1), để đơn giản và dễ dàng lấy mẫu. Phân phối prior được xác định độc lập với dữ liệu quan sát và duy trì cố định trong suốt quá trình huấn luyện. Nó cung cấp sự điều chỉnh và giúp ngăn chặn overfitting bằng cách áp đặt một ràng buộc trên không gian biến ẩn.
* Phân Phối Posterior: Phân phối posterior P(z∣x) đại diện cho niềm tin được cập nhật về các biến ẩn z sau khi quan sát dữ liệu x. Trong một VAE, phân phối posterior được xấp xỉ bằng cách sử dụng một mạng encoder chuyển đổi dữ liệu đầu vào thành các tham số của phân phối. Mục tiêu của mạng encoder là xấp xỉ phân phối posterior thực sự càng gần càng tốt, dựa trên dữ liệu quan sát được. Phân phối posterior phụ thuộc vào dữ liệu và thay đổi dựa trên dữ liệu đầu vào. Nó bắt chụp mối quan hệ giữa dữ liệu quan sát và các biến ẩn.
* Tóm lại, phân phối prior đại diện cho niềm tin trước khi quan sát dữ liệu, trong khi phân phối posterior đại diện cho niềm tin được cập nhật sau khi quan sát dữ liệu. Phân phối prior là cố định và độc lập với dữ liệu, trong khi phân phối posterior phụ thuộc vào dữ liệu và được học từ dữ liệu quan sát.

Nhiệm vụ của encoder là nắm bắt bản chất của hình ảnh đầu vào x và biểu diễn nó trong không gian tiềm ẩn bằng cách lập mô hình xác suất có điều kiện P(z∣x), được gọi là **posterior distribution**. Nó biểu thị xác suất của các biến tiềm ẩn z với dữ liệu được quan sát x. Tuy nhiên, việc xác định chính xác phân phối này khá phức tạp do mối quan hệ phức tạp giữa dữ liệu và các biến tiềm ẩn. Sự phức tạp này làm cho posterior trở nên khó giải quyết .

Theo định lý Bayes, chúng ta có thể tính phân phối sau P(z|x) như sau:



Encoder mô hình hóa khả năng P(x∣z), xác suất quan sát được hình ảnh x với các biến tiềm ẩn z. **Prior** **distribution** P(z) nắm bắt niềm tin hoặc giả định của chúng ta về các biến tiềm ẩn z trước khi quan sát bất kỳ hình ảnh x nào. Trong suy luận Bayes, chúng ta gọi nó là prior belief. Mặc dù chúng ta thường có thể cho rằng đó là một phân phối đơn giản, chẳng hạn như phân bố Gaussian, việc lựa chọn Prior distribution có thể có nhiều mặt:

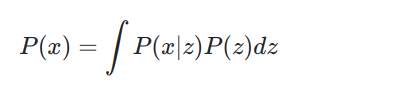
* Prior có thông tin: Được sử dụng khi chúng ta có thông tin biết trước hoặc niềm tin về một tham số cụ thể. Prior này phản ánh các kiến thức hoặc niềm tin được biết đến trước về tham số.
* Prior không thông tin hoặc phẳng: Được sử dụng khi thiếu thông tin biết trước, và mọi giá trị của tham số được gán trọng số bằng nhau.
* Prior đối nghịch (conjugate priors): Được chọn cho sự thuận tiện toán học, đảm bảo rằng phân phối posterior giữ nguyên dạng như prior.

Trong bối cảnh của VAE, sự lựa chọn prior (thường là standard normal distribution) được thúc đẩy bởi sự thuận tiện trong tính toán và mong muốn áp đặt các cấu trúc cụ thể lên không gian tiềm ẩn. Mặc dù lựa chọn này phù hợp với nguyên tắc Bayes về việc kết hợp các prior beliefs, nhưng trong trường hợp của VAE, nó thiên về “ những gì chúng ta muốn nó trở thành ” đối với hiệu quả của mô hình và các đặc tính mong muốn hơn là về “ những gì chúng ta tin nó là ”.

Nói tóm lại, chúng ta muốn phân phối các biến tiềm ẩn là phân phối standard normal distribution vì nó làm cho mô hình đơn giản và dễ lấy mẫu.

Quay trở lại công thức định lý Bayes, mẫu số P(x) biểu thị xác suất cận biên hoặc "evidence" (bằng chứng). Nó định lượng khả năng quan sát dữ liệu hình ảnh x mà không cần điều chỉnh bất kỳ giá trị cụ thể nào của z. Để xác định nó, chúng ta phải tính mật độ xác suất (hoặc khối lượng) của hình ảnh x cho mọi giá trị có thể hiểu được của z, sau đó lấy tích phân (hoặc tổng) trên tất cả các giá trị này.

Về mặt toán học, chúng ta có thể biểu diễn P(x) như sau:



Nếu chúng ta có thể tính toán chính xác P(x), chúng ta sẽ sử dụng P(z|x) để lấy mẫu các đặc điểm tiềm ẩn. Tuy nhiên, sự phức tạp đan xen của dữ liệu nhiều chiều, cấu trúc mô hình, không có phân phối xác định và nhu cầu tích hợp trên không gian tiềm ẩn khiến cho việc tính toán trực tiếp P(x) trên thực tế là không thể.

**Efficient Inference: Variational Inference**

Variational Inference (VI - Suy diễn biến phân) là một phương pháp được sử dụng để tính gần đúng các phân phối hậu nghiệm phức tạp, thường khó xử lý với các phân phối hậu nghiệm đơn giản hơn, dễ xử lý hơn.

Ý tưởng cốt lõi xoay quanh hai bước chính:

* Chọn một phân phối gần đúng : Chọn một họ phân phối, thường đơn giản hơn phân phối thực, để hoạt động như một phân phối gần đúng. Những phân phối này có các tham số mà chúng ta có thể điều chỉnh để làm cho phép tính gần đúng tốt hơn.
* Tối ưu hóa để giảm thiểu sự khác biệt : Điều chỉnh các tham số của phân bố gần đúng dựa trên dữ liệu được quan sát để làm cho nó càng gần với phân phối hậu nghiệm thực sự nhất có thể. Thước đo mức độ “gần gũi” thường là sự phân kỳ Kullback-Leibler (KL).

Trong quá trình suy diễn biến phân, bạn có thể tự hỏi tại sao các phân phối đơn giản có thể xấp xỉ một phân phối phức tạp một cách hiệu quả.

Trong khi phân phối hậu nghiệm thực sự có thể phức tạp và không thể tính toán trên toàn miền của nó, sức hấp dẫn của VI nằm ở sự cục bộ của nó. Thay vì cố gắng xấp xỉ toàn bộ phân phối, VI tập trung vào các khu vực liên quan đến dữ liệu quan sát. Bằng cách tập trung vào các khu vực cục bộ này, VI có thể sử dụng các phân phối đơn giản để xấp xỉ hành vi phức tạp của phân phối hậu nghiệm thực sự ở những nơi quan trọng nhất. Phương pháp chọn lọc này là lý do tại sao một phân phối dường như đơn giản có thể xấp xỉ một phân phối phức tạp hơn, và nó rất hiệu quả. Dưới đây là một phân tích dựa trên ẩn dụ có thể làm rõ hơn về tính cục bộ của suy diễn biến phân. Hãy tưởng tượng chúng ta đang cố gắng hiểu hình dạng của một dãy núi phức tạp với đỉnh núi, thung lũng và các địa hình phức tạp khác. Nếu chúng ta thử xấp xỉ toàn bộ, chúng ta sẽ cố gắng sử dụng một đường cong mượt mà duy nhất để bao quát toàn bộ dãy núi. Điều này sẽ rất khó khăn, yêu cầu tính toán mạnh mẽ, và có thể bỏ qua nhiều chi tiết. Hãy nghĩ đến một cách tiếp cận khác: Thay vì ánh xạ toàn bộ dãy núi, tập trung vào các phần nhỏ. Chúng ta sẽ xấp xỉ các đường cong cho những khu vực cục bộ này, bắt chước các chi tiết của chúng một cách chính xác. Theo thời gian, chúng ta nhằm mục tiêu xấp xỉ toàn bộ dãy núi một cách chính xác hơn bằng cách tổng hợp hiểu biết của chúng ta về nhiều khu vực như vậy. Chiến lược này của việc tập trung vào các khu vực hoặc vùng cụ thể, sau đó ghép chúng lại để hiểu rộng lớn hơn, tương ứng với nguyên tắc của sự xấp xỉ cục bộ trong VI.

**Efficient Learning: Deep Learning**

Hãy biểu thị posterior distribution gần đúng là Q ϕ (z∣x). Ở đây, ϕ đại diện cho các tham số mà chúng ta có thể điều chỉnh để fit Q ϕ (z∣x) với true posterior P(z|x).

Trong VAE, chúng ta sử dụng mạng thần kinh như một cách hiệu quả để tham số hóa và tối ưu hóa các phân phối gần đúng. Cụ thể, với một hình ảnh x, mạng nơ-ron đưa ra các tham số (như giá trị trung bình và phương sai) xác định phân phối Q ϕ (z∣x), từ đó chúng ta có thể lấy mẫu biến tiềm ẩn z.

Vì chúng ta muốn làm cho Q ϕ (z∣x) gần hơn với P(z∣x), nên chúng ta nên hướng tới việc giảm thiểu sự phân kỳ KL giữa hai phân phối mà chúng ta có thể đưa vào hàm mất mát của mình như một thuật ngữ chính quy hóa.

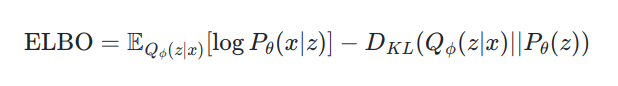
Câu hỏi trước mắt là: nếu P(z∣x) khó xử lý, làm thế nào chúng ta có thể tính được độ phân kỳ KL giữa P(z|x) và Q ϕ (z∣x)?

Câu trả lời là chúng ta không tính toán phân kỳ KL một cách trực tiếp. Thay vào đó, chúng ta tối đa hóa Evidence Lower Bound (ELBO) thu được từ sự phân kỳ KL. Bằng cách tối đa hóa ELBO, chúng ta ngầm giảm thiểu sự phân kỳ KL giữa phân bố gần đúng Q ϕ (z∣x) và P(z|x), mặc dù chúng ta không tính toán trực tiếp sự phân kỳ này.

**Evidence Lower Bound (ELBO): Deriving from KL Divergence**

Trong VAE, chúng ta sử dụng encoder để ước tính quá trình tạo P(x∣z) và chúng ta có quyền kiểm soát prior P(z). Vì điều này, chúng ta sử dụng ký hiệu P θ để biểu thị các phân phối được tham số hóa này. Cụ thể, P θ (x∣z) biểu thị phân bố xác suất của việc quan sát hình ảnh x với các biến tiềm ẩn z được mô hình hóa bởi decoder và P θ (z) biểu thị Prior distribution đã chọn của chúng ta cho các biến tiềm ẩn.

Biến đổi các kiểu con đà điểu thì ra công thức này :v

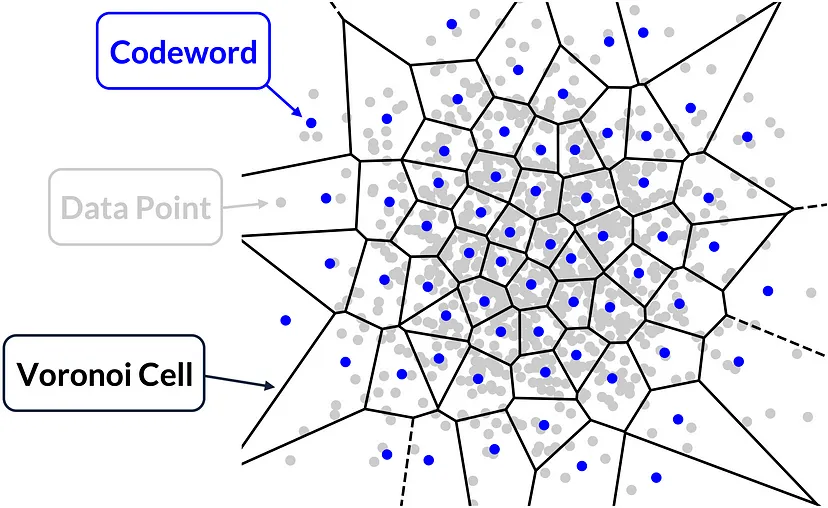


Công thức ELBO không bao gồm true posterior P(z|x) và chúng ta có thể sử dụng chúng để phục vụ các mục đích kép sau:

* Maximizing Data Likelihood : Thuật ngữ E Q ϕ (z∣x) [log P θ (x|z)] biểu thị khả năng log-likelihood dự kiến ​​​​của dữ liệu hình ảnh x với các biến tiềm ẩn được mã hóa bởi encoder z. Bằng cách tối đa hóa thuật ngữ này, chúng tai mong muốn đảm bảo rằng dữ liệu được xây dựng lại (từ decoder) càng gần với dữ liệu ảnh gốc x càng tốt.
* Regularizing the Latent Space : Thuật ngữ D KL (Q ϕ (z∣x)∣∣P θ (z)) hoạt động như một công cụ chính quy. Nó đảm bảo rằng sự phân bố của các biến tiềm ẩn z, như được mô hình hóa bởi decoder, không sai lệch quá nhiều so với Prior distribution mà chúng ta đã chọn. Thuật ngữ này khuyến khích không gian tiềm ẩn duy trì cấu trúc mong muốn, cho phép chúng ta lấy mẫu z để tạo ra hình ảnh mới.

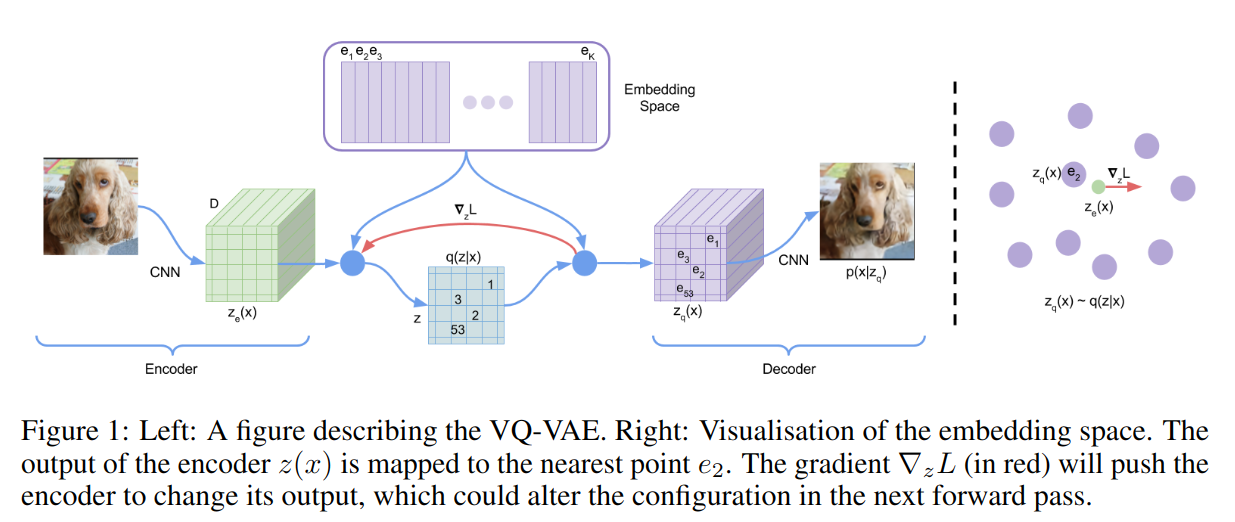
Bằng cách tối đa hóa ELBO, chúng ta đạt được hai mục tiêu sau: chúng ta đảm bảo rằng VAE tái tạo lại dữ liệu một cách chính xác trong khi vẫn duy trì không gian tiềm ẩn có cấu trúc.

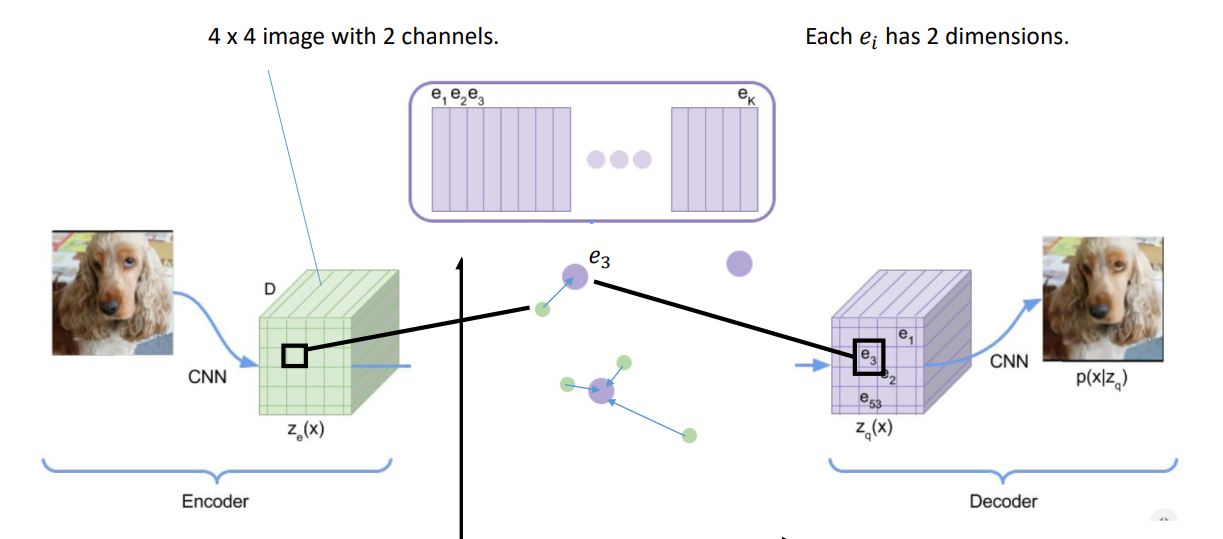
**2.2 Vector Quantization**

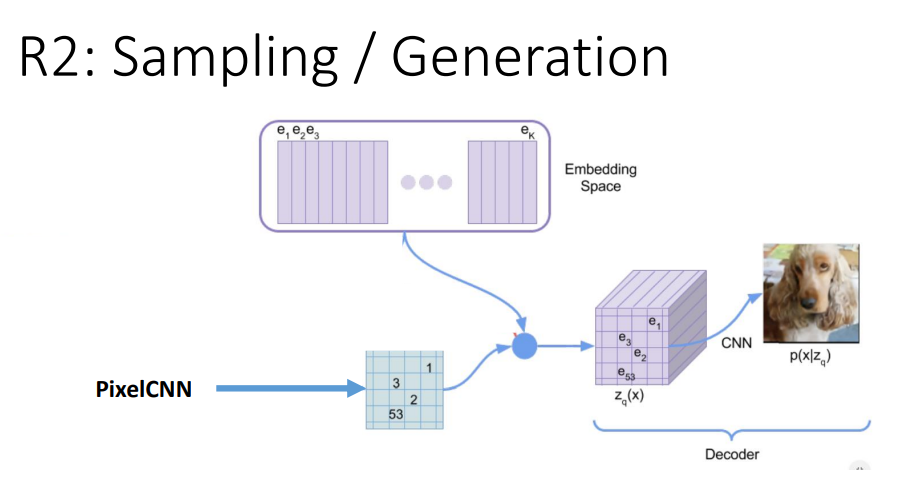
****

Lượng tử hóa vectơ (VQ) là một kỹ thuật nén dữ liệu tương tự như thuật toán k-means có thể mô hình hóa bất kỳ phân phối dữ liệu nào. Lượng tử hóa vectơ đã được sử dụng trong nhiều ứng dụng cho dữ liệu giọng nói, hình ảnh và video, chẳng hạn như tạo hình ảnh, mã hóa giọng nói và âm thanh, chuyển đổi giọng nói, tạo nhạc, và tổng hợp văn bản thành giọng nói. Đối với quy trình VQ, chúng ta yêu cầu codebook bao gồm một số codeword. Áp dụng VQ trên một điểm dữ liệu (các chấm màu xám) có nghĩa là ánh xạ nó tới codeword gần nhất (các chấm màu xanh), tức là thay thế giá trị của điểm dữ liệu bằng giá trị codeword gần nhất. Mỗi ô voronoi (đường màu đen) chứa một codeword sao cho tất cả các điểm dữ liệu nằm trong ô đó sẽ được ánh xạ tới codeword đó, vì nó là codeword gần nhất với các điểm dữ liệu nằm trong ô voronoi đó.

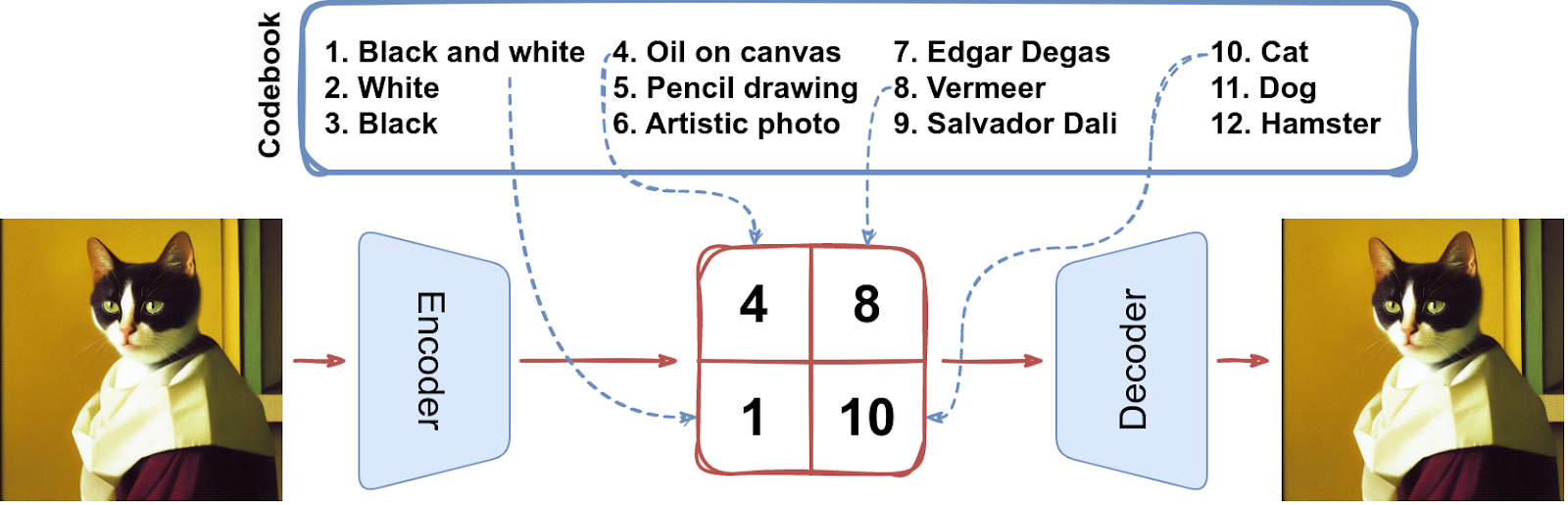
1. **VQ-VAE**

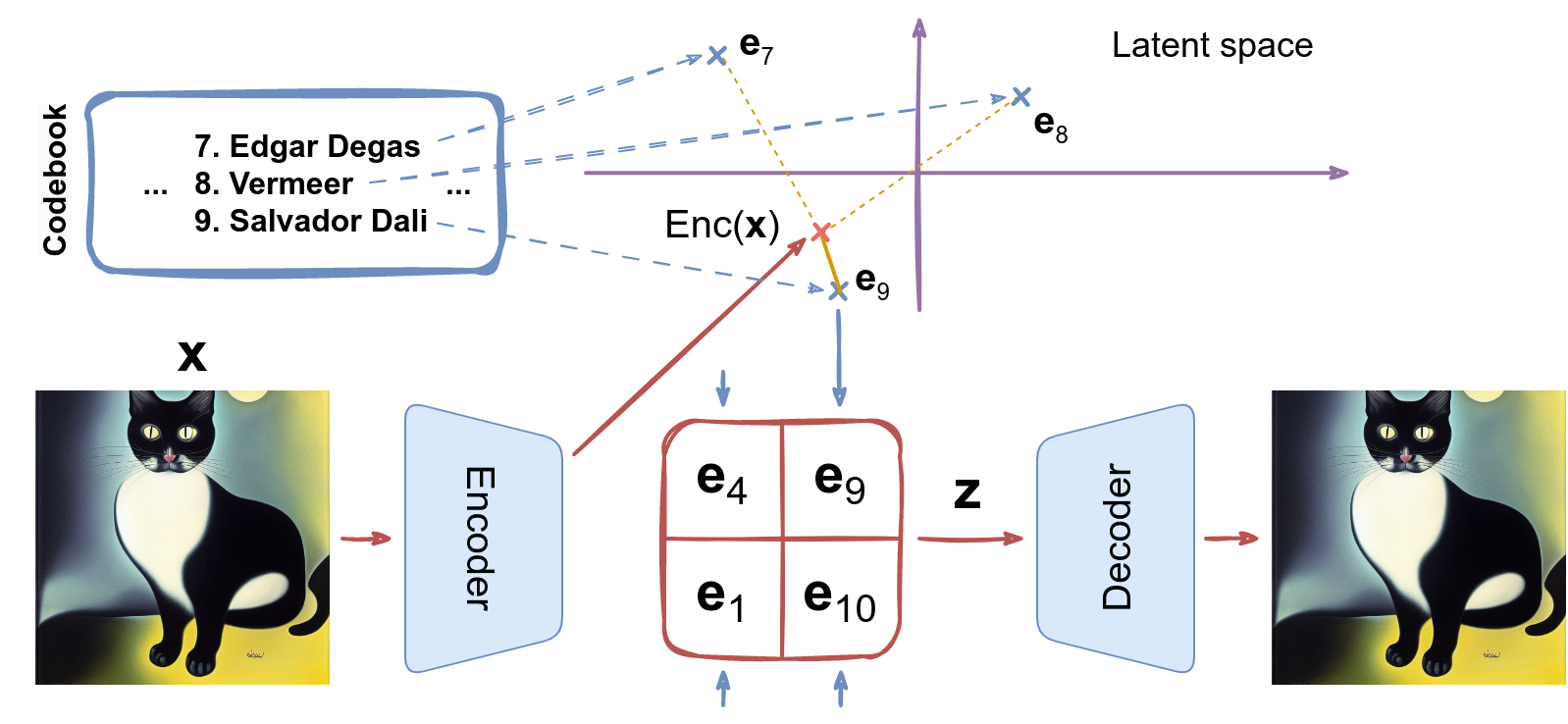
****

****

****

* 1. **Discrete Latent variables**



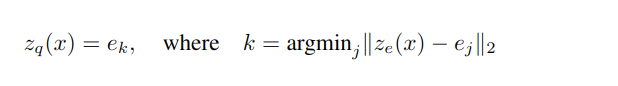


* Trong mô hình này, chúng ta xác định một không gian nhúng tiềm ẩn e với K vector nhúng, mỗi vector có kích thước D (tức là một không gian nhúng phân loại K-cách, trong đó K là số lượng phân loại và D là số chiều của mỗi vector nhúng e i ​ ). Cụ thể, ta có K vector nhúng e i ​ có kích thước D, với i=1,2,...,K. Quá trình đầu tiên là đưa đầu vào x qua một Encoder để tạo ra đầu ra z e ​ (x), là một vector trong không gian biến phân. Sau đó, các biến phân rời rạc z được tính bằng cách sử dụng phương pháp tìm kiếm hàng xóm gần nhất, sử dụng không gian nhúng chia sẻ e như mô tả trong phương trình 1. Đầu vào cho Decoder là vector nhúng tương ứng e k ​ như mô tả trong phương trình 2. Chúng ta có thể xem quá trình tính toán này như một Encoder tự động thông thường với một hàm phi tuyến cụ thể chuyển các biến phân thành vector nhúng 1 trong K-cách. Tổng hợp các tham số của mô hình bao gồm tham số của Encoder, Decoder và không gian nhúng e. Để đơn giản, chúng ta sử dụng một biến ngẫu nhiên duy nhất z để đại diện cho các biến phân rời rạc trong phần này, tuy nhiên, thực tế, đối với âm thanh, hình ảnh và video, chúng ta thực sự trích xuất các không gian đặc trưng nhúng 1D, 2D và 3D tương ứng.
* Phân phối hậu nghiệm q(z|x) phân loại được xác định như sau:



Ở đây, z e ​ (x) là đầu ra của mạng Encoder. Chúng ta xem mô hình này như một VAE trong đó chúng ta có thể giới hạn logp(x) bằng ELBO (Evidence Lower Bound). Phân phối đề xuất q(z=k∣x) của chúng ta là xác định, và bằng cách xác định một phân phối đồng đều đơn giản trên z, chúng ta thu được một hằng số độ lệch KL bằng logK.

* Biểu diễn z e ​ (x) được chuyển qua đầu vào giới hạn rời rạc tiếp theo trước khi ánh xạ vào phần tử gần nhất của nhúng e như mô tả trong phương trình 1 và 2.



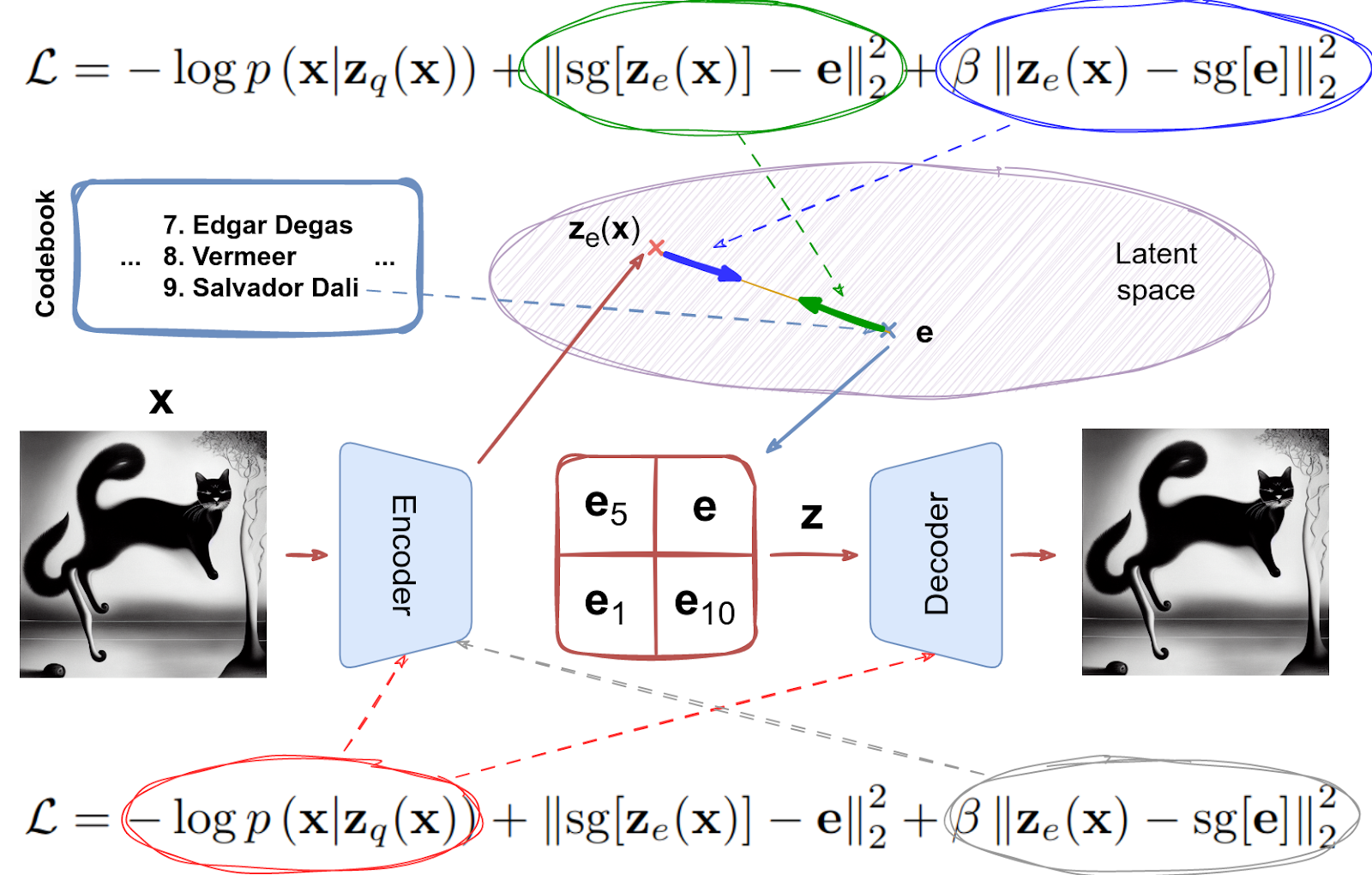
* 1. **Learning**
* Lưu ý rằng không có gradient thực sự được xác định cho phương trình 2. Tuy nhiên, chúng ta cần tính toán đạo hàm để huấn luyện mô hình. Để làm điều này, chúng ta sử dụng một phương pháp gần đúng tương tự như ước lượng trực tiếp (straight-through estimator). Ý tưởng cơ bản của phương pháp này là chúng ta giả sử rằng việc ánh xạ ngược từ đầu ra của Decoder đến đầu ra của Encoder là một phép toán liên tục. Do đó, chúng ta có thể sao chép đạo hàm từ đầu ra của Decoder (z q ​ (x) đến đầu ra của Encoder ( z e ​ (x)). Điều này giúp chúng ta tính được đạo hàm của hàm mất mát liên quan đến z e ​ (x). Người ta cũng có thể sử dụng gradient con thông qua phép toán lượng tử hóa, nhưng công cụ ước tính đơn giản này hoạt động tốt cho các thí nghiệm ban đầu trong bài báo này.
* Trong quá trình tính toán xuôi, zq (x) (phương trình 2) vector nhúng gần nhất được truyền tới Decoder và trong quá trình truyền ngược, gradient ∇z L được truyền không thay đổi đến Encoder. Do biểu diễn đầu ra của Encoder và đầu vào của Decoder có chung không gian D chiều, nên các gradient chứa thông tin hữu ích về cách Encoder phải thay đổi đầu ra của nó để giảm reconstruction loss.
* Phương trình 3 chỉ định hàm loss tổng thể:



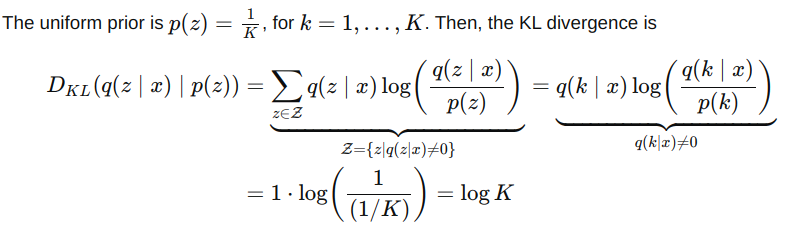
=> **logp(x∣z q ​ (x))**: Đây là phần của reconstruction loss, trong đó p(x∣z q ​ (x)) đại diện cho xác suất có điều kiện của dữ liệu x được tái tạo từ biến latents được giới hạn rời rạc z q ​ (x). Điều này đo lường sự tương đồng giữa dữ liệu gốc và dữ liệu được tái tạo bởi mô hình.

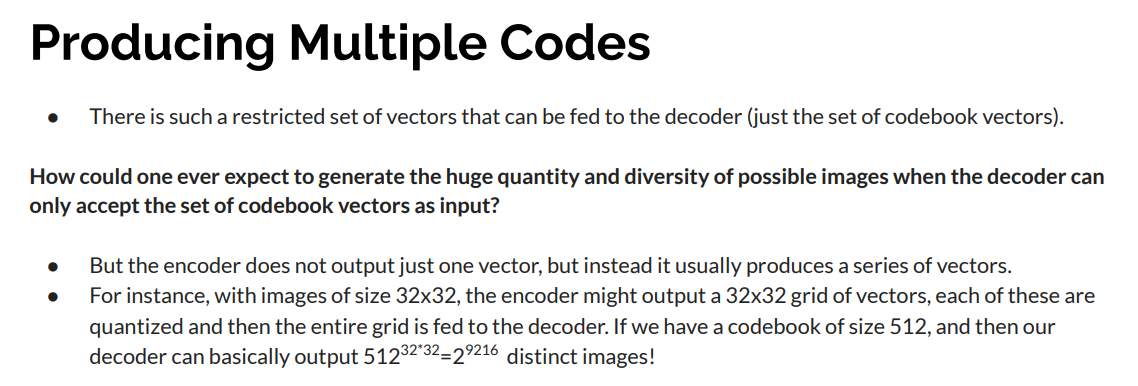
=> **∥sg[ze(x)]−e∥** ​ : Đây là thành phần của hàm mất mát để đảm bảo rằng vector biểu diễn đầu ra của Encoder ze(x) gần với các vector nhúng e. sg là viết tắt của toán tử stopgradients, g[ze(x)] đại diện cho bản sao của ze(x), nhưng nó không tham gia vào việc cập nhật tham số trong quá trình lan truyền ngược. Mục tiêu của phần này là để tạo ra một không gian biểu diễn gần với không gian nhúng được định nghĩa trước (e).

=> **β∥ze(x)−sg[e]∥** ​ : Đây là thành phần của hàm mất mát cam kết (commitment loss). Điều này giúp đảm bảo rằng Encoder sẽ cam kết vào các giá trị latents z, và đầu ra của nó không tăng lên không kiểm soát. sg[e] đại diện cho bản sao của e, và nó không tham gia vào quá trình cập nhật tham số trong quá trình lan truyền ngược. Mục tiêu là giảm thiểu khoảng cách này, đồng nghĩa với việc Encoder phải chọn các giá trị latents gần với giá trị nhúng e, và không thay đổi quá nhiều giữa các lần chạy. β là hệ số để điều chỉnh độ ảnh hưởng của thành phần cam kết này đối với hàm mất mát tổng.

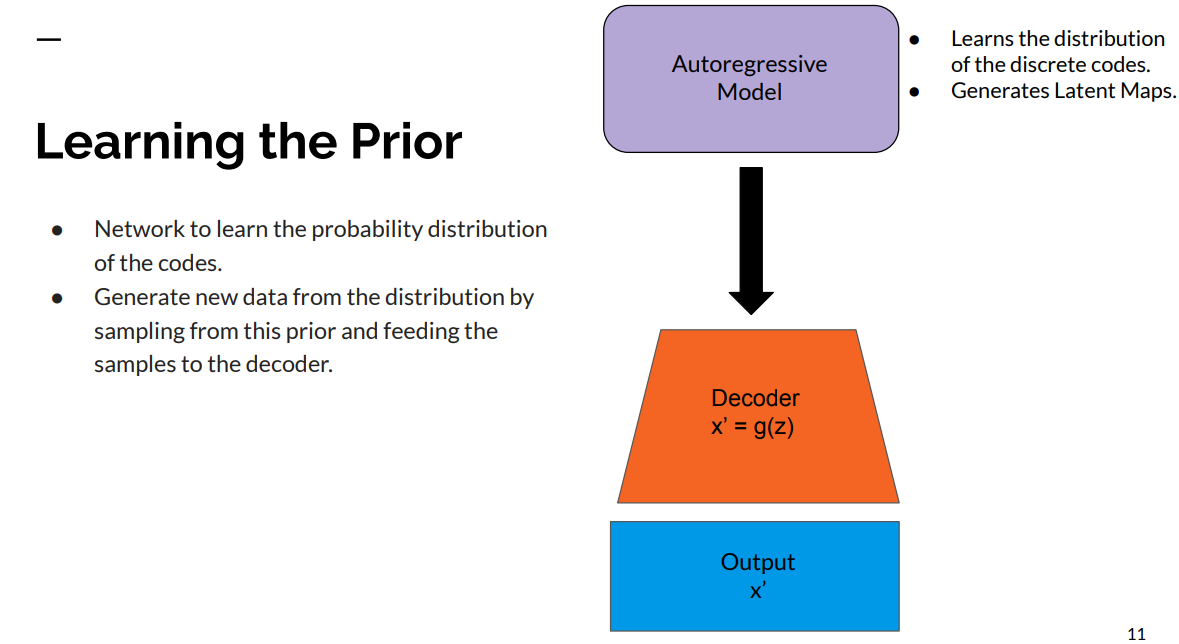


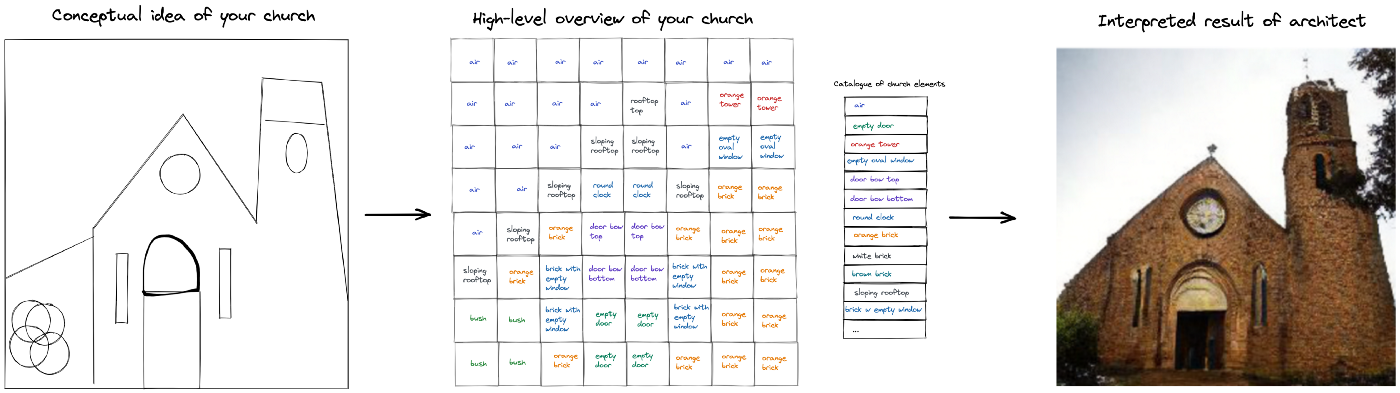
* Decoder chỉ tối ưu hóa thành phần loss đầu tiên, Encoder tối ưu hóa thành phần loss đầu tiên và cuối cùng, đồng thời các phần nhúng được tối ưu hóa theo phần loss ở giữa. Tác giả nhận thấy thuật toán thu được khá chắc chắn đối với β, vì kết quả không thay đổi đối với các giá trị của β nằm trong khoảng từ 0,1 đến 2,0. Tác giả sử dụng β = 0,25 trong tất cả các thử nghiệm của mình, mặc dù nhìn chung điều này phụ thuộc vào quy mô reconstruction loss. Vì chúng ta giả sử một uniform prior z nên thuật ngữ KL thường xuất hiện trong ELBO là hằng số w.r.t. các tham số Encoder và do đó có thể được bỏ qua khi huấn luyện.





* 1. **Prior**

****

****

* Ý tưởng đầu tiên là sử dụng một phân phối tiền lập để lấy mẫu từ các biến latents, tương tự như Variational Autoencoder. Vì chúng ta đang xử lý các biến rời rạc, nên chúng ta có thể sử dụng một uniform prior trên các mã rời rạc, để mỗi mã có cùng xác suất được chọn. Tuy nhiên, trong thực tế, điều này dẫn đến các mẫu không đồng nhất và chất lượng thấp. Nhìn vào ảnh trên, bạn không thể chỉ đặt một cửa sổ ngẫu nhiên vào không gian và một mái nhà dưới cánh cửa của bạn và mong đợi kiến trúc sư tạo ra một cái gì đó hợp lý. Điều này đến từ việc trong phân phối dữ liệu gốc, các mã không độc lập và 'sự phù hợp' của một mã phụ thuộc vào mã nào đã tồn tại. Chúng ta cần một cách biểu diễn tốt hơn về phân phối các mã rời rạc để tạo ra các hình ảnh mạch lạc hơn..
* Để giải quyết vấn đề lấy mẫu này, bài báo ban đầu đề xuất tìm hiểu prior distribution thay vì sử dụng uniform prior cố định.
* Để mô hình hóa các xác suất được hệ số hóa này, bài báo đã sử dụng mô hình pixelCNN. Mô hình này sử dụng masked sliding window, xem xét các biến xung quanh để dự đoán biến tiếp theo. So với simple uniform prior, phương pháp này cải thiện đáng kể khả năng tạo ra hình ảnh nhất quán mới bằng cách tìm hiểu các mã xuất hiện cùng nhau.
* Tổng thể:
  + Huấn luyện một VQ-VAE: Huấn luyện mô hình VQ-VAE trên bộ dữ liệu của bạn để học một biểu diễn rời rạc của dữ liệu đầu vào.
  + Mã hóa Bộ dữ liệu: Xử lý toàn bộ bộ dữ liệu huấn luyện thông qua Encoder của VQ-VAE đã được huấn luyện. Điều này sẽ dẫn đến các biểu diễn đã được mã hóa cho mỗi điểm dữ liệu, được biểu diễn bằng các chỉ số của các mục trong codebook.
  + Tạo một Bộ dữ liệu Mới: Tổ chức các biểu diễn đã được mã hóa này thành một bộ dữ liệu mới. Mỗi điểm dữ liệu trong bộ dữ liệu mới này tương ứng với biểu diễn đã được mã hóa của một hình ảnh đầu vào.
  + Huấn luyện một Mô hình Tự hồi quy: Huấn luyện một mô hình tự hồi quy (như PixelCNN) trên bộ dữ liệu mới này. Thay vì dự đoán giá trị pixel, mô hình tự hồi quy dự đoán các chỉ số của các mục trong codebook. Bạn có thể chọn đưa vào là chỉ số hoặc các vector trong codebook, với các mục tiêu là chỉ số.
  + Tạo mẫu: Khi mô hình tự hồi quy được huấn luyện, bạn có thể tạo ra các chuỗi từ đó. Các chuỗi này đại diện cho các biểu diễn đã được mã hóa của các hình ảnh. Truyền các chuỗi này qua Decoder của VQ-VAE để tái tạo lại các hình ảnh.
  + Bằng cách làm việc với các biểu diễn đã được mã hóa, thu nhỏ, mô hình tự hồi quy sẽ dễ huấn luyện hơn, và bạn có thể sử dụng các mô hình nhỏ hơn và ít phức tạp hơn. Sự tích hợp này cho phép chúng ta tận dụng các điểm mạnh của cả VQ-VAE và mô hình tự hồi quy để tạo ra các mẫu chất lượng cao.

1. **Experimental & Results**
   1. **Comparison with continuous variables**

* So sánh hiệu suất của ba mô hình: VAE (Variational Autoencoder) với biến liên tục, VQ-VAE (Vector Quantized Variational Autoencoder), và VIMCO (Variational Inference for Monte Carlo Objectives) với prior là Gaussian độc lập và phân phối phân loại. Các kết quả hiệu suất được đo bằng bits/dim (bits trên mỗi kích thước pixel) và so sánh như sau:

- VAE: 4.51 bits/dim

- VQ-VAE: 4.67 bits/dim

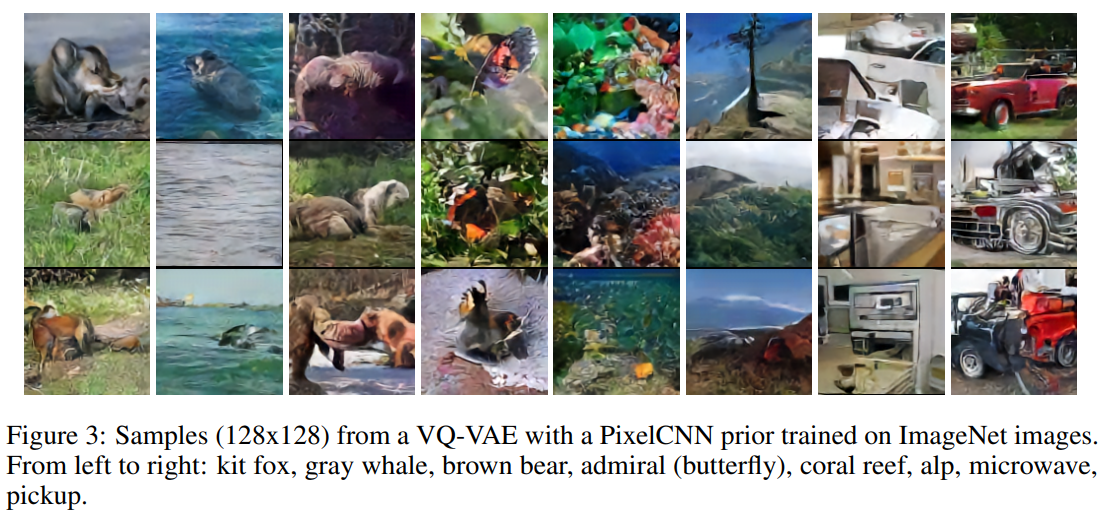
- VIMCO: 5.14 bits/dim

Điều đáng chú ý ở đây là VQ-VAE, mặc dù sử dụng không gian latent rời rạc, nhưng vẫn đạt được hiệu suất gần bằng với VAE sử dụng không gian latent liên tục. Điều này cho thấy rằng việc sử dụng không gian latent rời rạc trong VQ-VAE không làm giảm quá nhiều hiệu suất so với sử dụng không gian latent liên tục trong VAE. Điều này làm cho VQ-VAE trở thành một lựa chọn hấp dẫn khi cần sử dụng biểu diễn latent rời rạc để tạo ra các mô hình sinh hoặc phân tích dữ liệu.

* Bits/dim là một đơn vị được sử dụng để đo lường mức độ hiệu quả của mô hình sinh dữ liệu (như Autoencoder, Generative Adversarial Networks, và các biến thể của chúng). Đơn vị này biểu thị số lượng bits (thông tin) mỗi kích thước pixel (hoặc mỗi chiều không gian pixel) cần để mô hình sinh ra hoặc tái tạo dữ liệu.
* Mức độ bits/dim càng thấp thì mô hình càng hiệu quả, vì nó yêu cầu ít thông tin hơn để biểu diễn mỗi pixel. Trong ngữ cảnh của mô hình sinh dữ liệu, việc giảm bits/dim thường được coi là một mục tiêu quan trọng, vì nó đồng nghĩa với việc mô hình sinh ra dữ liệu giống như dữ liệu thực tế hơn.
  1. **Images**
* Trong Hình 2, ảnh bên phải là ảnh được chuyển đổi từ ảnh trái sang không gian tiềm ẩn z và sau đó được khôi phục bằng Decoder VQ-VAE. Độ phân giải và độ chi tiết bị giảm đi phần nào, nhưng bạn có thể thấy rằng phần tổng thể vẫn được nắm bắt mà không làm mất thông tin quan trọng mặc dù kích thước đã giảm đi rất nhiều.



* Một số hình ảnh được tạo từ VQ-VAE bằng PixelCNN được hiển thị trong hình 3.



1. **References**

<https://kikaben.com/vae-2013/>

<https://stats.stackexchange.com/questions/595049/whats-the-role-of-the-commitment-loss-in-vq-vae>

<https://ai.stackexchange.com/questions/17203/what-is-the-input-for-the-prior-model-of-vq-vae>

<https://synthesis.ai/2023/03/21/generative-ai-ii-discrete-latent-spaces/>

-------------------Ending Mệt vãi ---------------------