Efficient Text-to-Image Generation via Structured Discrete Prediction

Sadeep Jayasumana

Based on:

MarkovGen: Structured Prediction for Efficient Text-to-Image Generation, Google Research, CVPR 2024.
Rethinking FID: Towards a Better Evaluation Metric for Image Generation, Google Research, CVPR 2024.

Image Generation Methods

- Diffusion models:
  - Stable Diffusion, Imagen, Dall-E 2
- Transformer-based models:
  - Parti, Muse

Image Generation Methods

- Diffusion models:
  - Stable Diffusion, Imagen, Dall-E 2
- Transformer-based models:
  - Parti, Muse

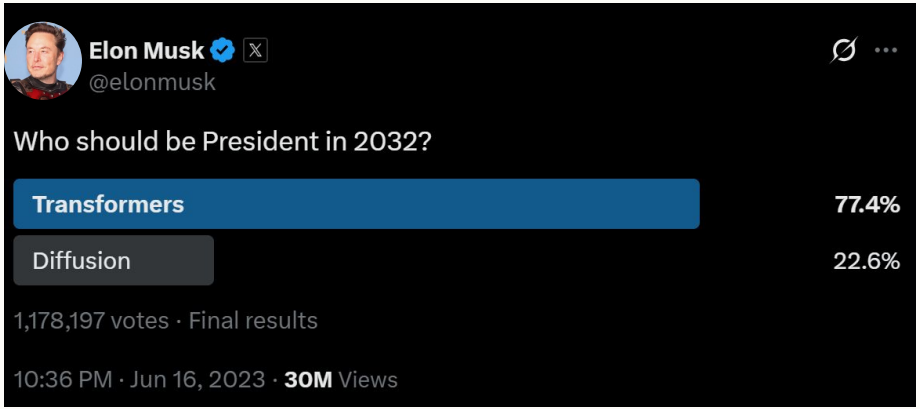


Image Tokenization

256 x 256 x 3

Tokenizer

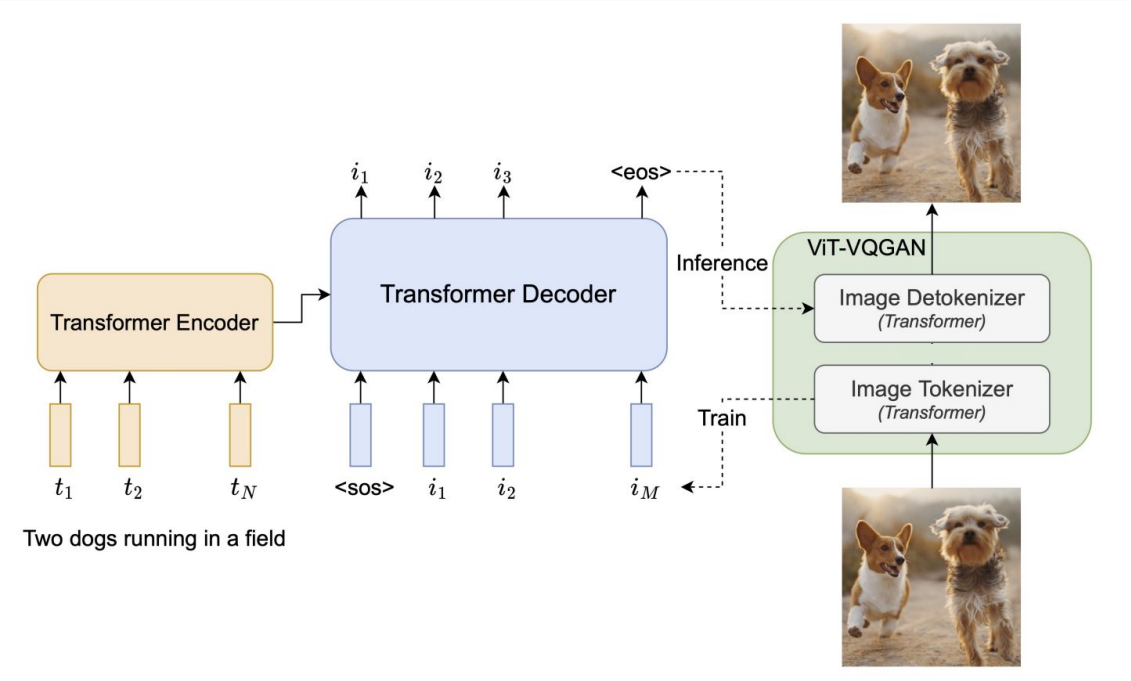
9	28	15	78
19	36	27	32
8	56	68	71
96	85	49	82

16 x 16

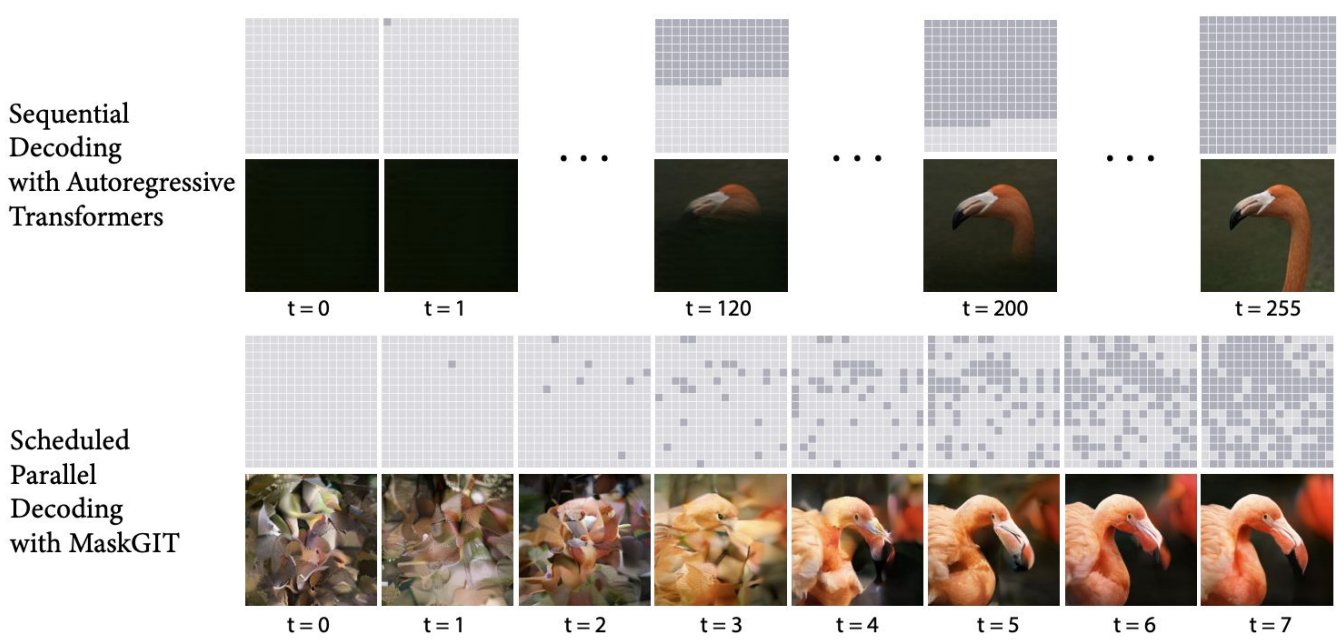
Detokenizer

256 x 256 x 3

Parti (Autoregressive)

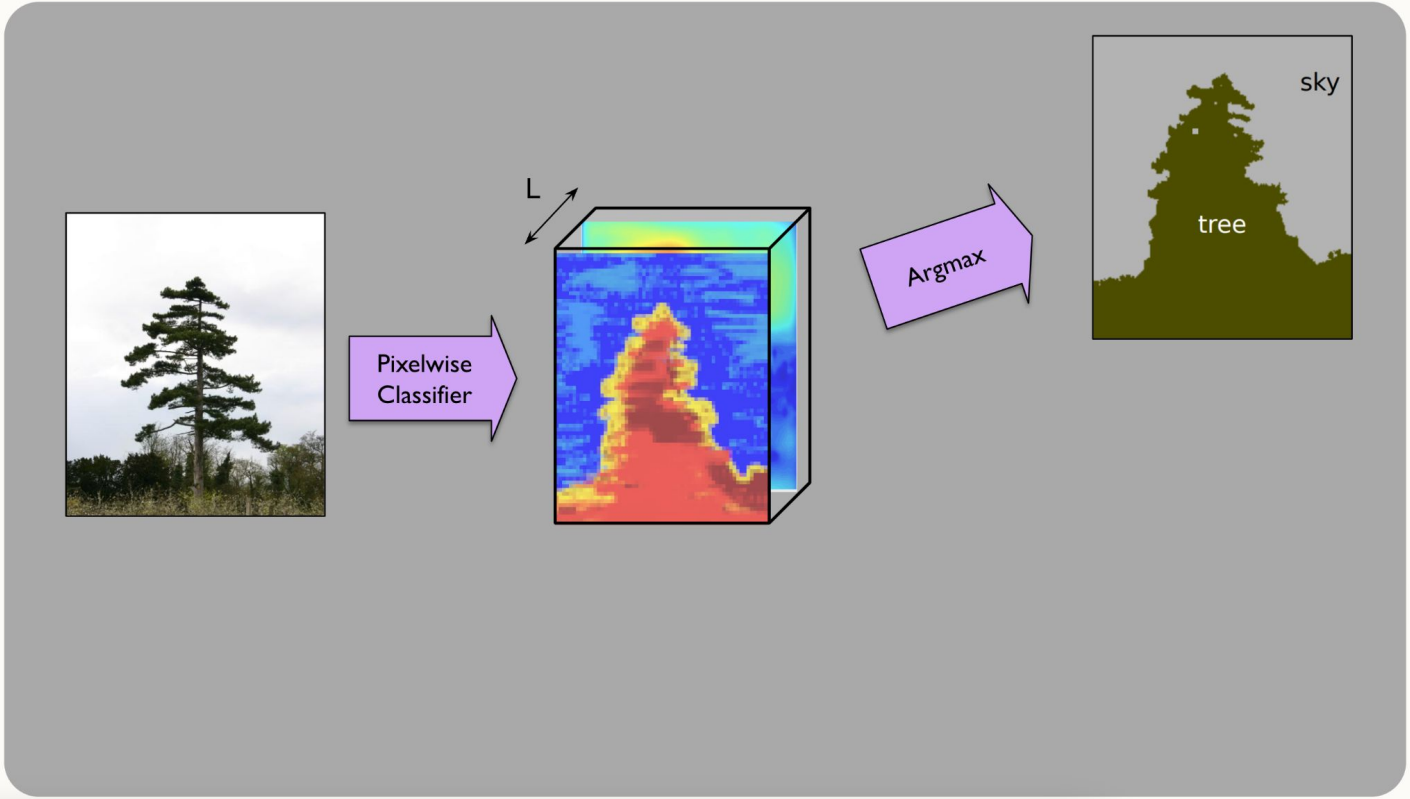


Muse (Parallel Decoding)

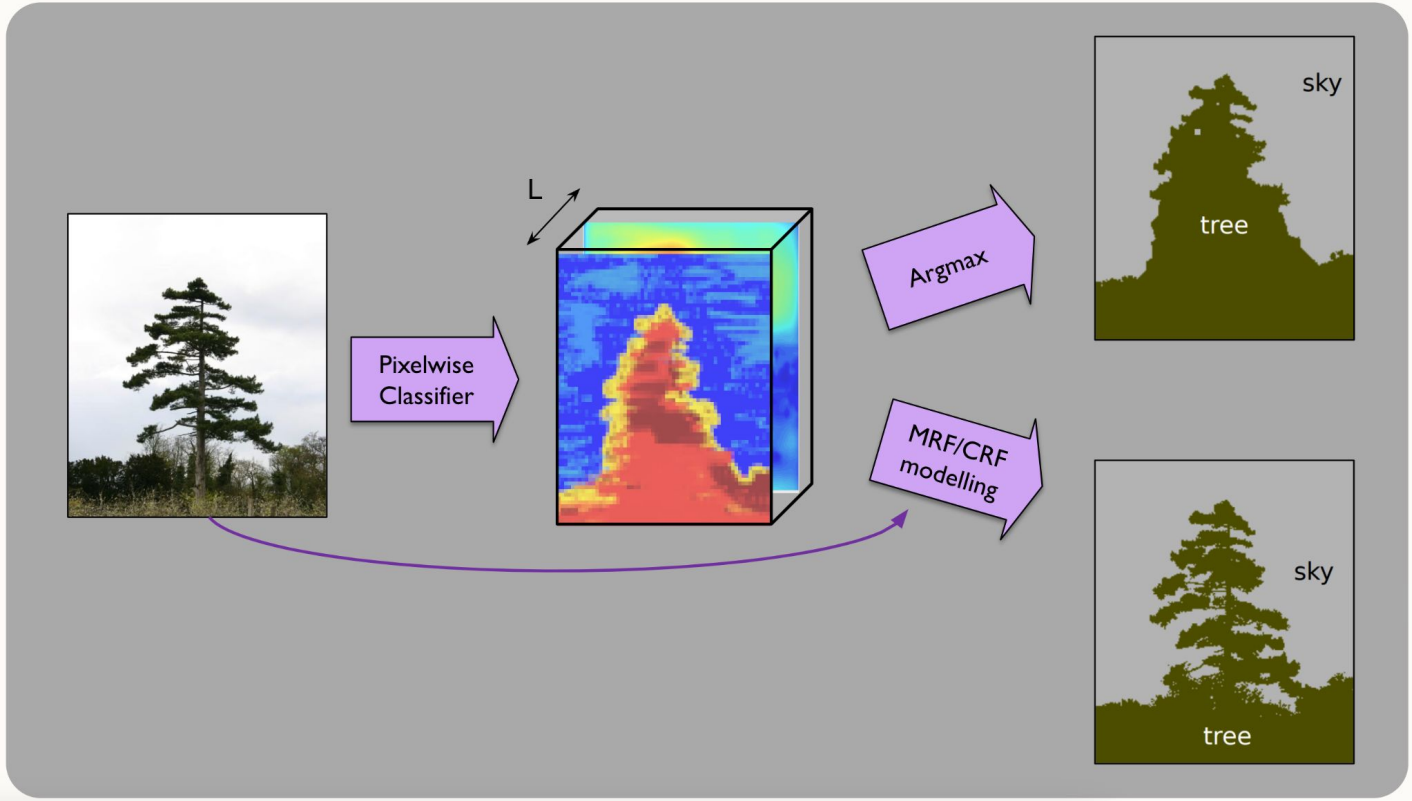


- The Muse 3B model is 10x faster than Parti/Imagen 3B on TPUv4.

Markov Random Fields (MRFs)

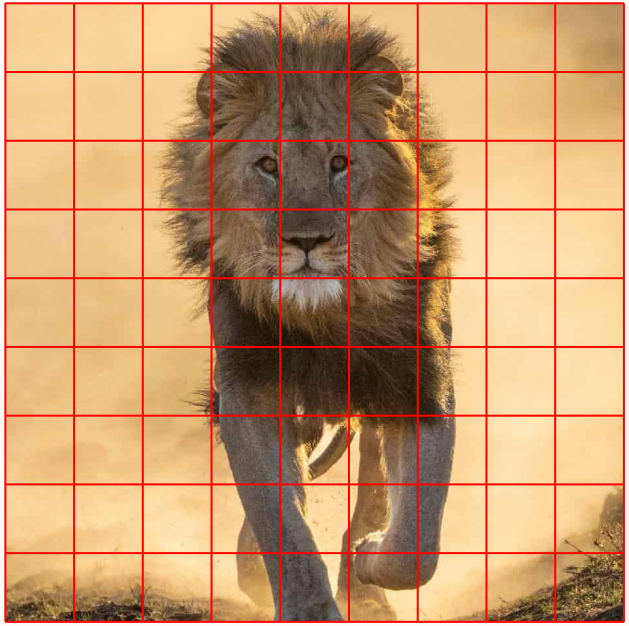


Markov Random Fields (MRFs)

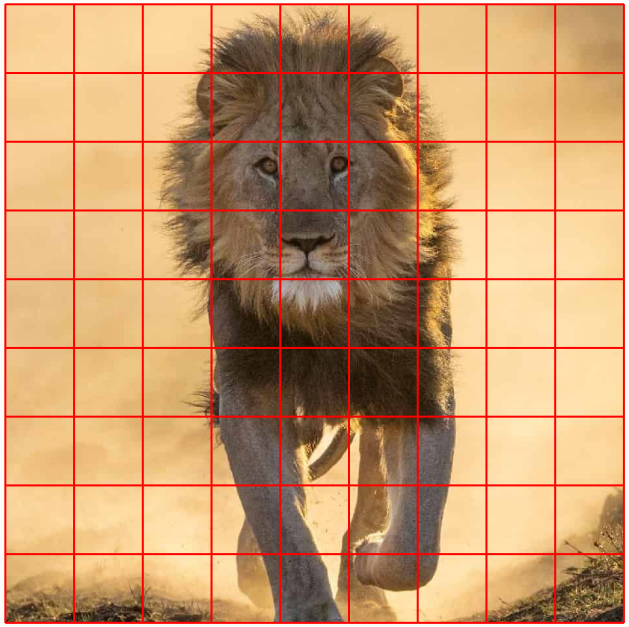


Motivation

Motivation

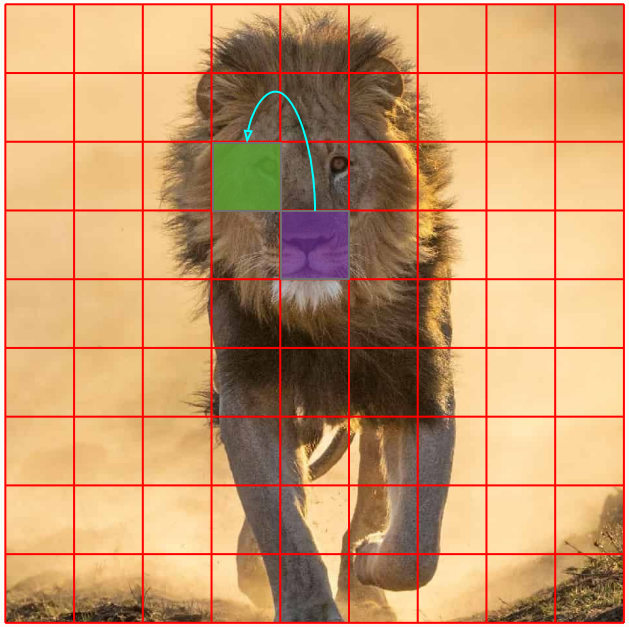


Motivation



- 8192 tokens in the vocab.
- Number of permutations:
  - 2x2 patch: $O(10^{15})$
  - 3x3 patch: $O(10^{35})$
  - 16x16 patch: $O(10^{1002})$
- Only a small subset of token arrangements will be “valid”.
- Highly confident tokens should be able to influence nearby tokens

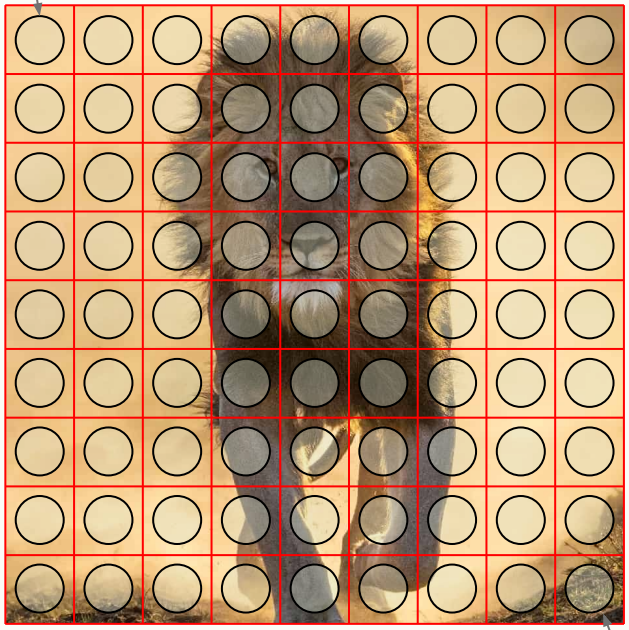
Motivation



- 8192 tokens in the vocab.
- Number of permutations:
  - 2x2 patch: $O(10^{15})$
  - 3x3 patch: $O(10^{35})$
  - 16x16 patch: $O(10^{1002})$
- Only a small subset of token arrangements will be “valid”.
- Highly confident tokens should be able to influence nearby tokens

Markov Random Fields (MRFs)

$X_1 \in \{l_1, l_2, ..., l_L\}$



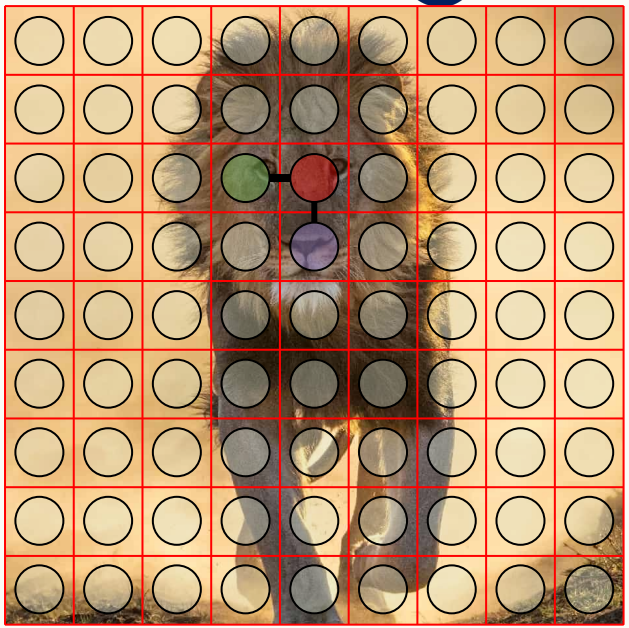
$X_N \in \{t_1, t_2, ..., t_V\}$

- Define a discrete random variable X_i at each cell *i*.
- Connect the random variables to form a random field.
- An assignment to the random field $X_1, X_2, \ldots, X_N \Rightarrow$ an image.

Markov Random Fields (MRFs)

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N) = P(\mathbf{X} = \mathbf{x})$$

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp(-E(\mathbf{x}))$$



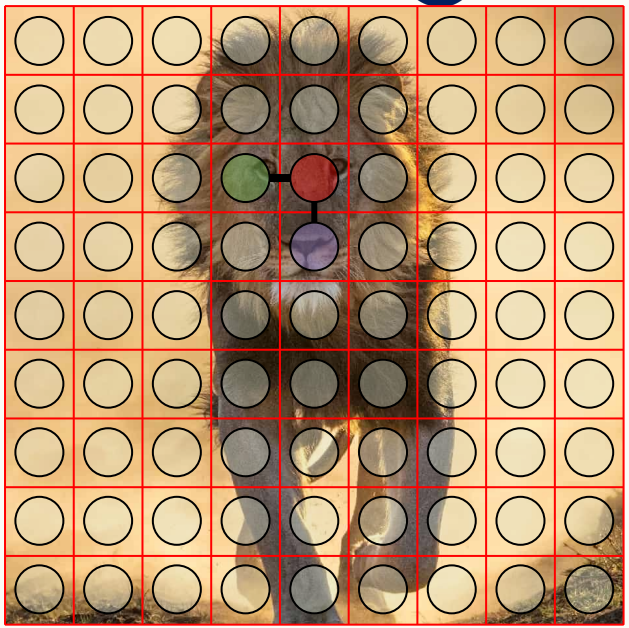
- Maximize $P(\mathbf{X} = \mathbf{x}) \implies$ Minimize $E(\mathbf{X} = \mathbf{x})$
- We now need to define $E(\mathbf{x})$ such that a photorealistic image will have low $E(\mathbf{x})$.

Model Formulation

$$E(\mathbf{x}) = \text{unary\_cost} + \text{pairwise\_cost}$$

Unary Cost

- $\text{cost}(X_i = l) = ?$
- You pay a penalty if your label doesn't agree with the classifier.



Model Formulation

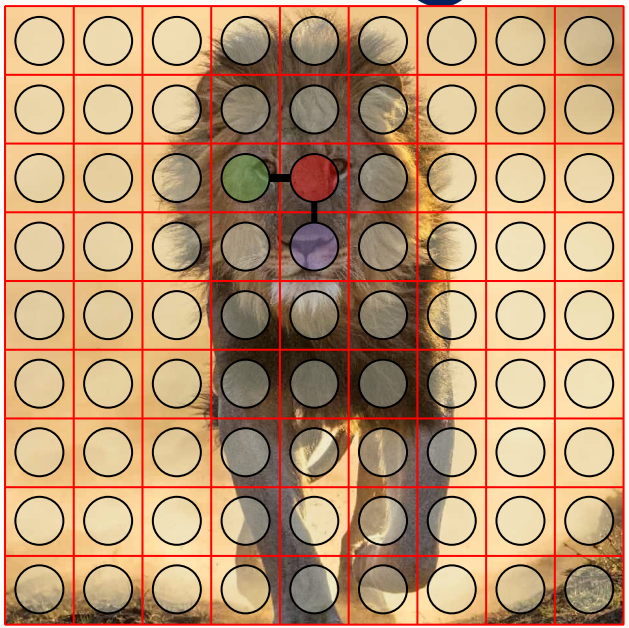
$$E(\mathbf{x}) = \text{unary\_cost} + \text{pairwise\_cost}$$

Unary Cost

- $\text{cost}(X_i = l) = ?$
- You pay a penalty if your label doesn't agree with the classifier.

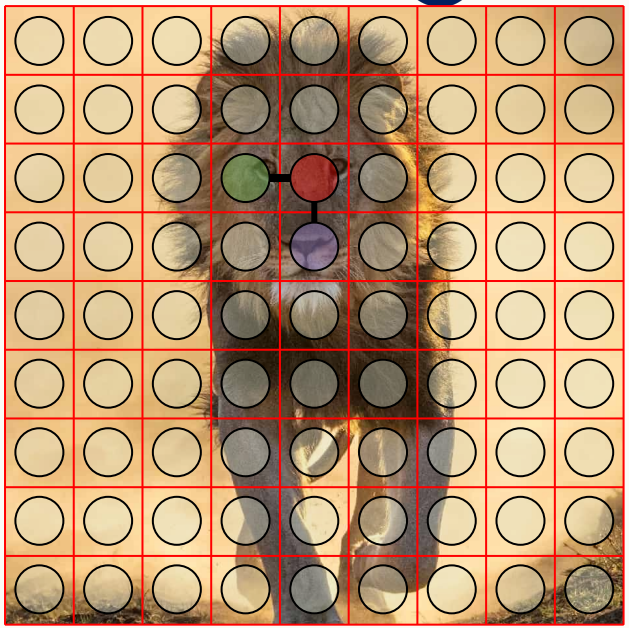
Pairwise cost

- $\text{cost}(X_i = l', X_j = l'') = ?$
- You pay a penalty if you assign *"incompatible"* labels to two *"neighboring"* pixels.



Model Formulation

$$E(\mathbf{x}) = \text{unary\_cost} + \text{pairwise\_cost}$$



Unary Cost

- $\text{cost}(X_i = l) = ?$
- You pay a penalty if your label doesn't agree with the classifier.

Pairwise cost

- $\text{cost}(X_i = l', X_j = l'') = ?$
- You pay a penalty if you assign *"incompatible"* labels to two *"neighboring"* pixels.

$$\text{cost}(X_i = l) = -\text{logit}_i(l)$$

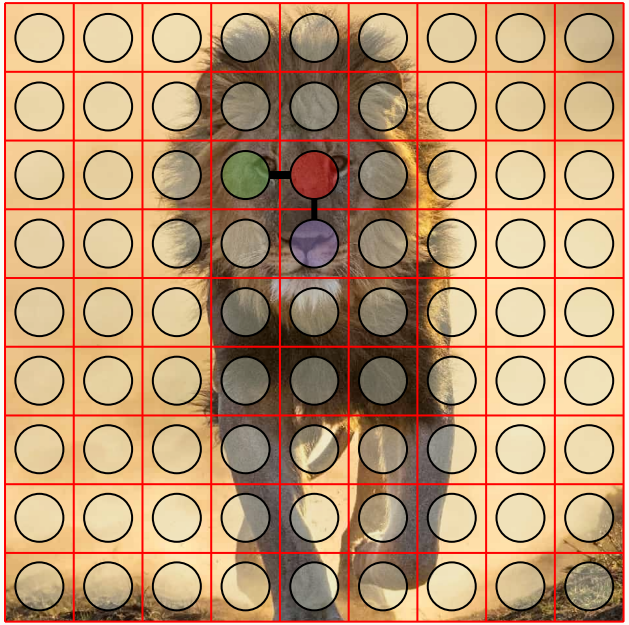
$$\text{cost}(X_i = l', X_j = l'') = -c(l', l'')s(i, j)$$

Difference Compared to Semantic Segmentation

- The graph is truly fully-connected.
- Spatial relationships are not fixed.
- Label compatibilities are not fixed.

$$\text{cost}(X_i = l) = -\text{logit}_\text{i}(l)$$

$$\text{cost}(X_i = l', X_j = l'') = -c(l', l'')s(i, j)$$



Inference Algorithm

$$E(\mathbf{x}|\mathbf{I}) = \sum_i \text{unary}(x_i) + \sum_{i>j} \text{pairwise}(x_i, x_j)$$

Inference Algorithm

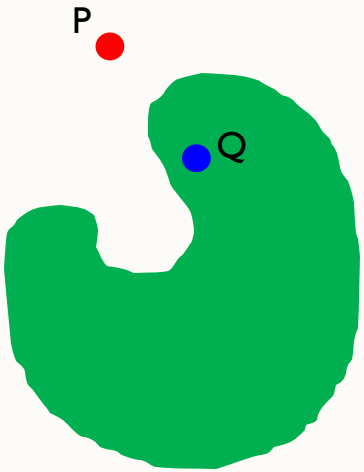
$$E(\mathbf{x}|\mathbf{I}) = \sum_i \text{unary}(x_i) + \sum_{i>j} \text{pairwise}(x_i, x_j)$$

$$\frac{1}{Z(\mathbf{I})} \exp\left(-E(\mathbf{x}|\mathbf{I})\right) = P(\mathbf{X} = \mathbf{x}|\mathbf{I}) \approx \prod_{i=1}^{N} Q_i(x_i)$$

Inference Algorithm

$$E(\mathbf{x}|\mathbf{I}) = \sum_i \text{unary}(x_i) + \sum_{i>j} \text{pairwise}(x_i, x_j)$$

$$\frac{1}{Z(\mathbf{I})} \exp\left(-E(\mathbf{x}|\mathbf{I})\right) = P(\mathbf{X} = \mathbf{x}|\mathbf{I}) \approx \prod_{i=1}^{N} Q_i(x_i)$$

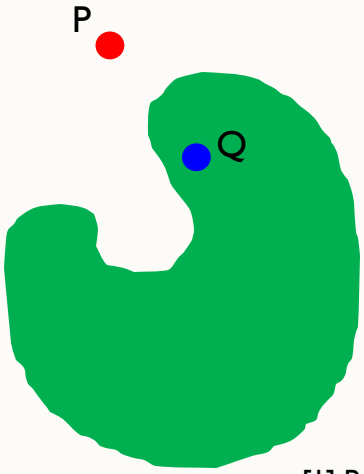


Inference Algorithm

$$E(\mathbf{x}|\mathbf{I}) = \sum_i \text{unary}(x_i) + \sum_{i>j} \text{pairwise}(x_i, x_j)$$

$$\frac{1}{Z(\mathbf{I})} \exp\left(-E(\mathbf{x}|\mathbf{I})\right) = P(\mathbf{X} = \mathbf{x}|\mathbf{I}) \approx \prod_{i=1}^{N} Q_i(x_i)$$

$$D_{\text{KL}}(Q\|P) = \mathbb{E}_Q[\log(Q(\mathbf{x})) - \log(P(\mathbf{x}))]$$



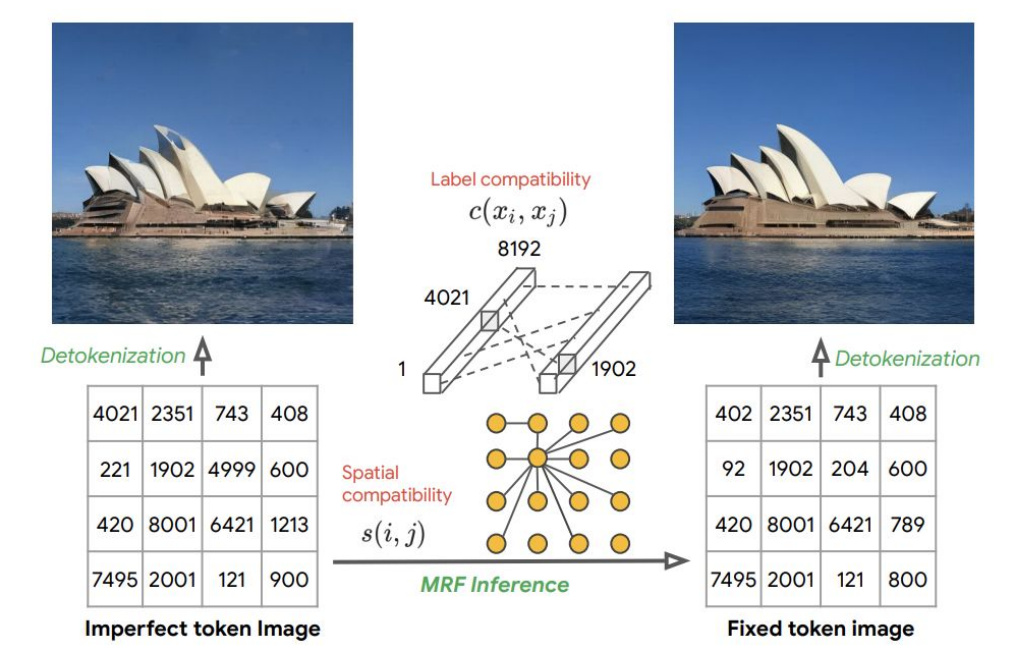
[1] P. Krähenbühl and V. Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials, NeurIPS , 2011
[2] D. Koller and N. Friedman. Probabilistic Graphical Models: Principles and Techniques. MIT Press, 2009

Inference Algorithm

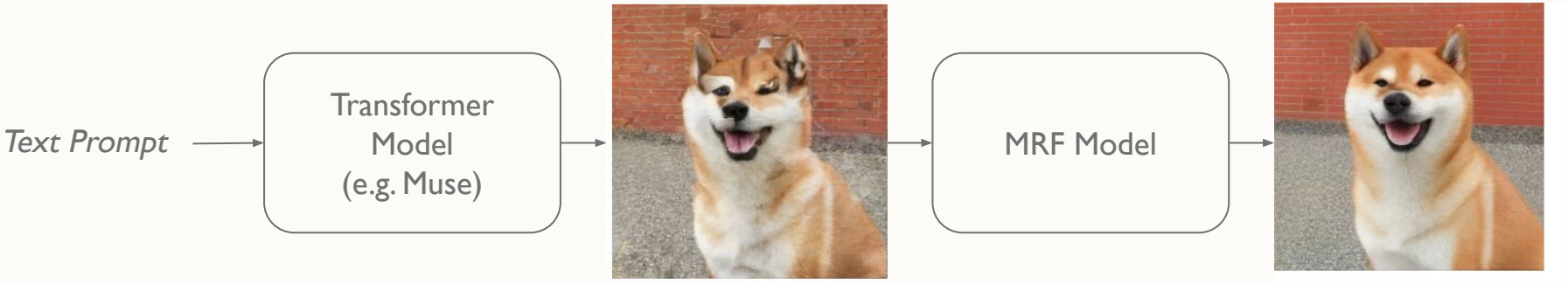
Algorithm 1 Inference Algorithm

$Q_i(k) \leftarrow \text{softmax}(f_i(k)), \forall(i,k)$
for `num_iterations` **do**
  $Q_i(k) \leftarrow \sum_{j=1}^{n} \mathbf{W^s}_{ij} Q_j(k), \forall(i,k)$
  $Q_i(k) \leftarrow \sum_{k'=1}^{V} \mathbf{W^c}_{kk'} Q_i(k'), \forall(i,k)$
  $Q_i(k) \leftarrow Q_i(k) + f_i(k), \forall(i,k)$
  $Q_i(k) \leftarrow \text{softmax}(Q_i)(k), \forall(i,k)$
end for
return Q

MRFs for Fast Image Generation



MRFs for Fast Image Generation



- The heavy-lifting is done here.
- Bulky, slow model

- Fixes the incompatible tokens
- Light-weight and super fast

Speeding Up Inference with MRFs

Model	Time (ms)
Muse base (single step)	10.40
Muse super-resolution (single step)	24.00
MRF inference on base	0.29
MRF inference on super-resolution	0.29
Detokenizer	0.15
Muse	442.05
MarkovGen (ours)	281.03

Qualitative Results

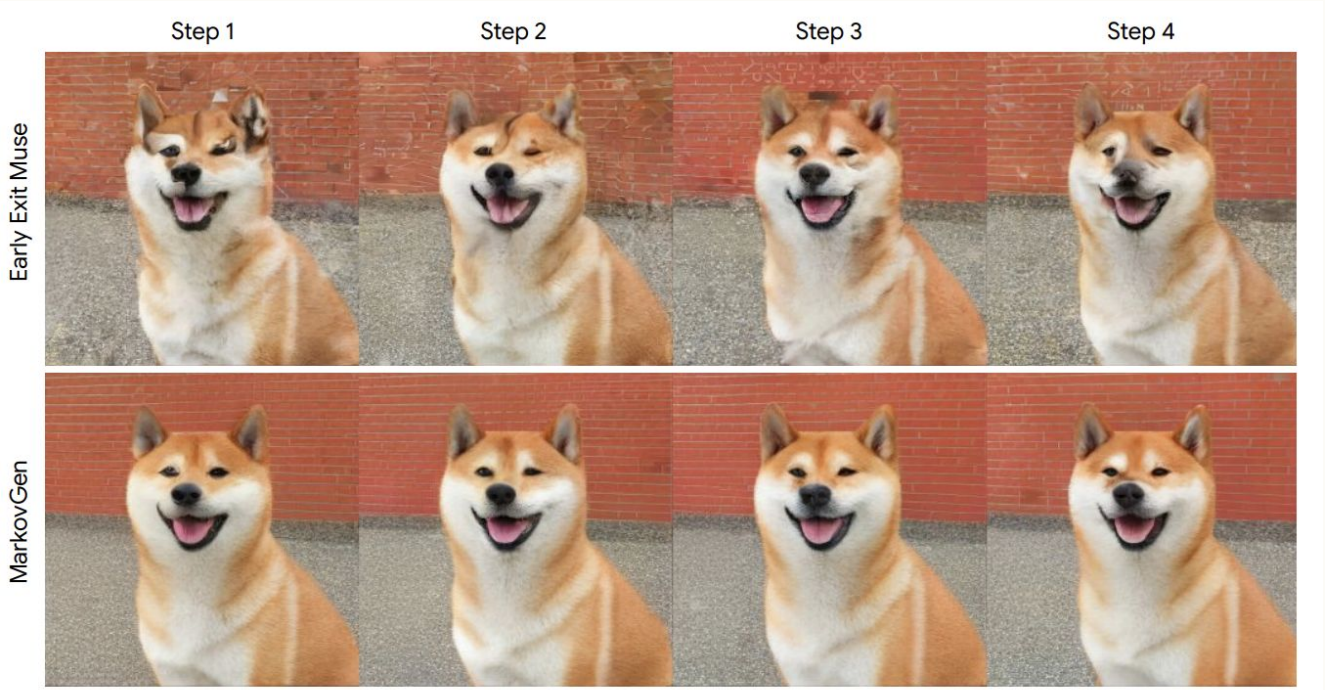


Figure 4. *The first four steps of the Muse super-resolution model without (top) and with (bottom) the application of the MarkovGen MRF model. Note that the MRF fixes complex object structures such as the dog's face as well as texture-inconsistencies in areas such as the brick wall. MarkovGen generates good looking high quality images starting from the first step.*

The finale of a fireworks display

An oil painting of two rabbits in the style of American Gothic, wearing the same clothes as in the original.

A robot painted as graffiti on a brick wall. a sidewalk is in front of the wall, and grass is growing out of cracks in the concrete.

A set of 2x2 emoji icons with happy, angry, surprised and sobbing faces. The emoji icons look like pandas. All of the pandas are wearing colorful sunglasses.

Figure 6. *Within each set of three, MarkovGen (right) speeds up Muse (left) by* $1.5\times$ *and improves image quality. A similar speed up by only reducing the step count with early exit Muse (middle) results in a significant loss of quality.*

Qualitative Results

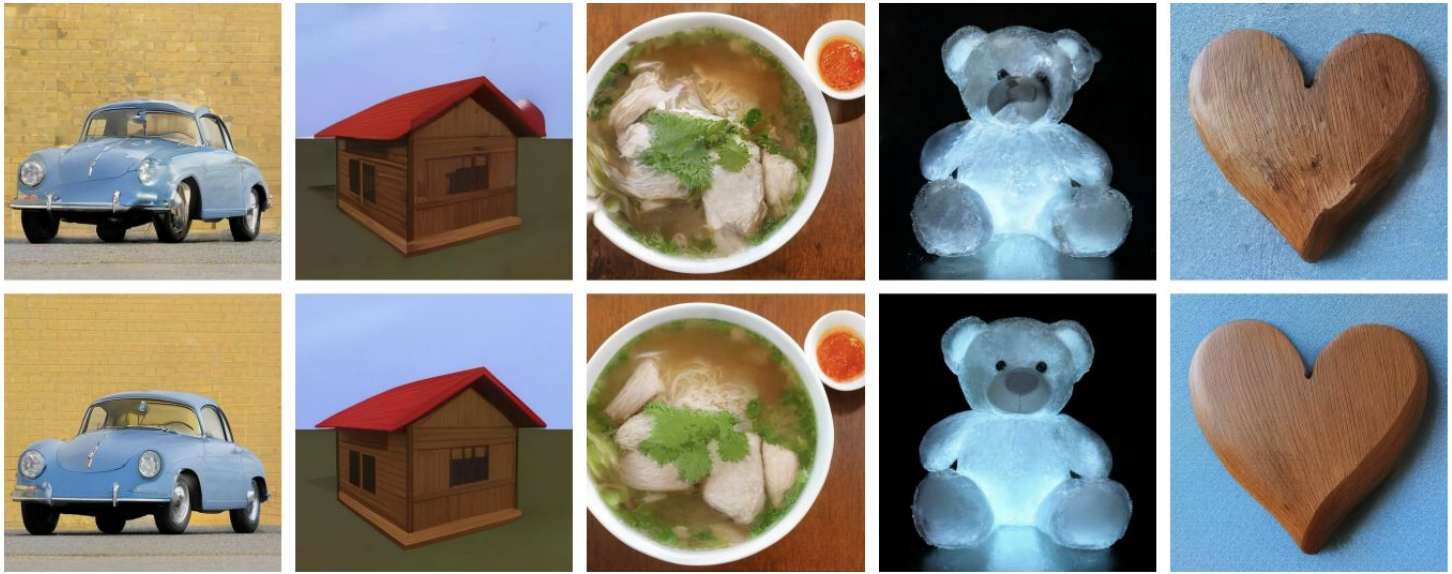


Figure 7. *Example generations of the Early Exit Muse super-resolution model running for 3 (out of 8) steps (top) and the MarkovGen model after the application of the* MRF *model (bottom). We observe a significant reduction in visual artifacts, e.g., in the brick wall behind the car. We further see key improvements to complex object structures such as the blue car and the teddy bear's face.*

Quantitative Results

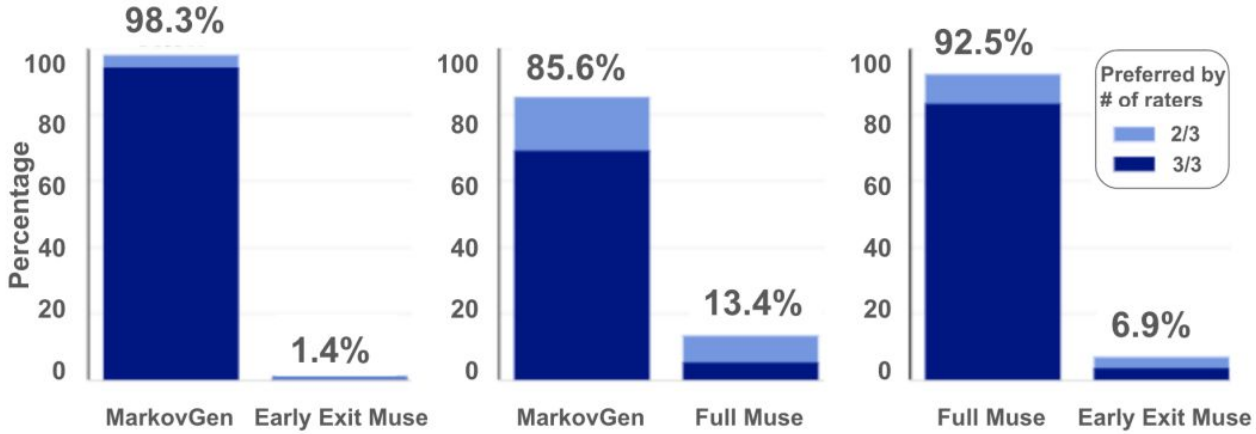


Figure 5. *Percentage of prompts for which human raters prefer images by a given model in a side-by-side comparison. We observe that human raters strongly prefer the images generated by MarkovGen over those of both early exit Muse (left) and even the more expensive and slower full Muse model (center).*

Part II:

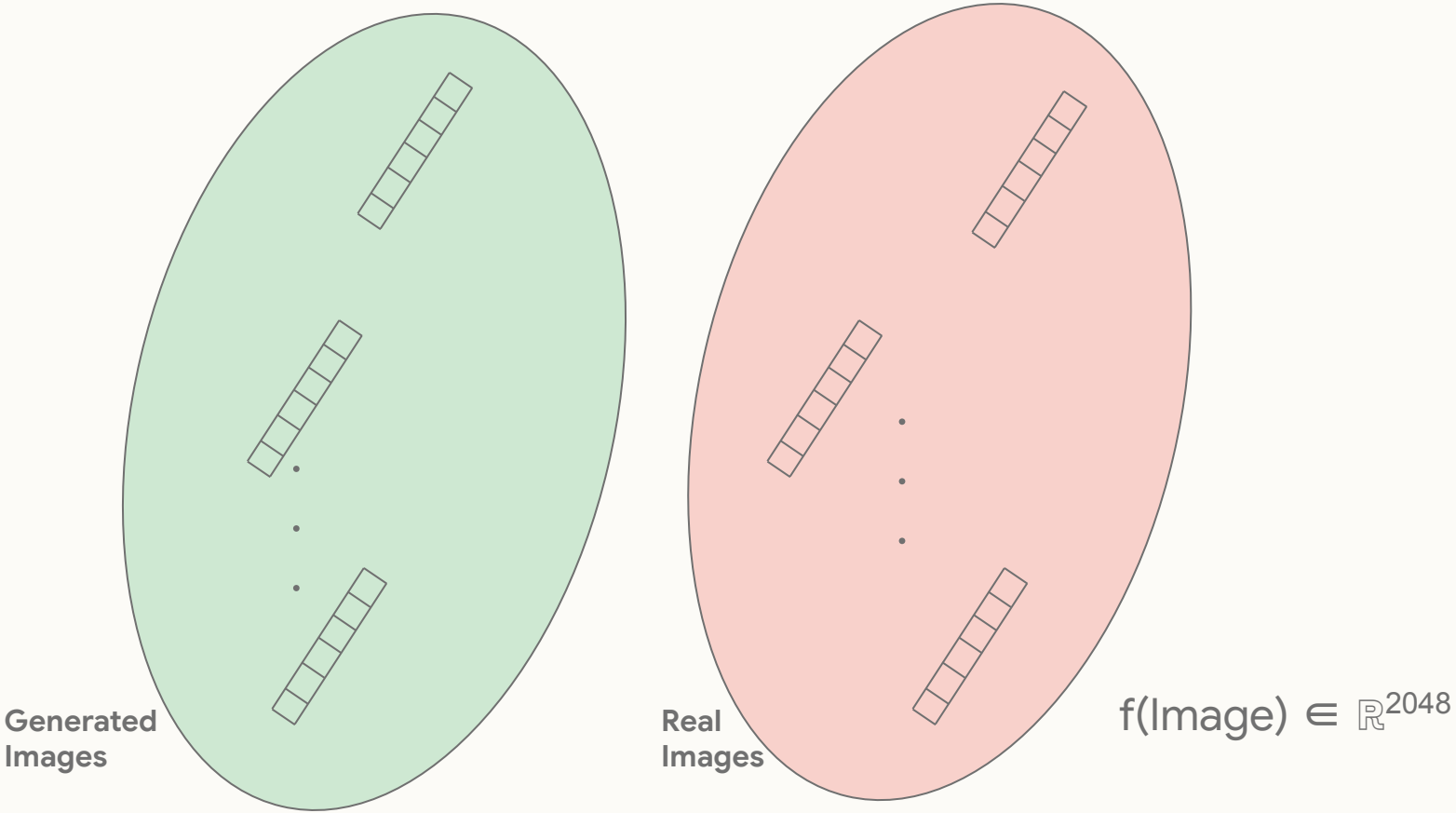
Rethinking FID as an Evaluation Metric for Image Generation

Comparing Two Image Distributions

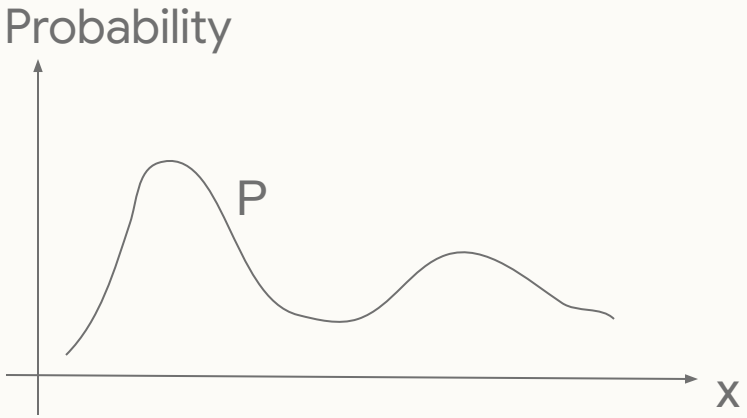
Generated Images

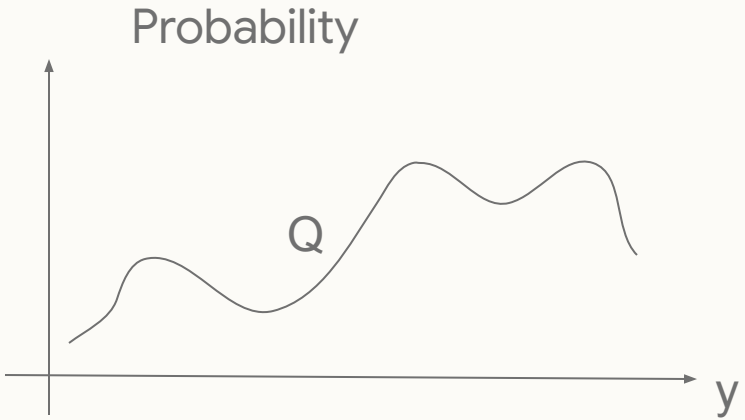
Real Images

Comparing Two Distributions

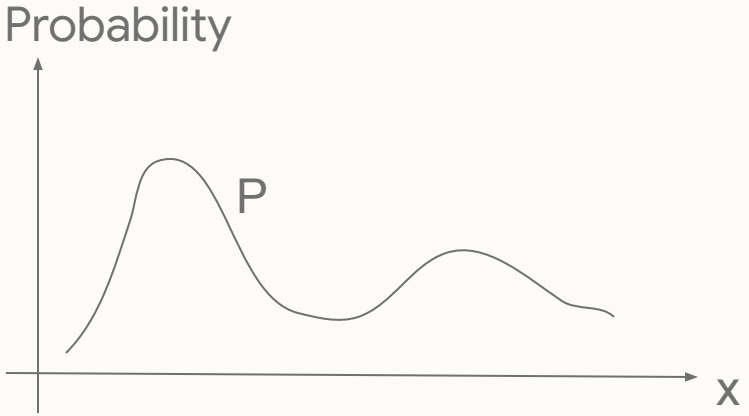


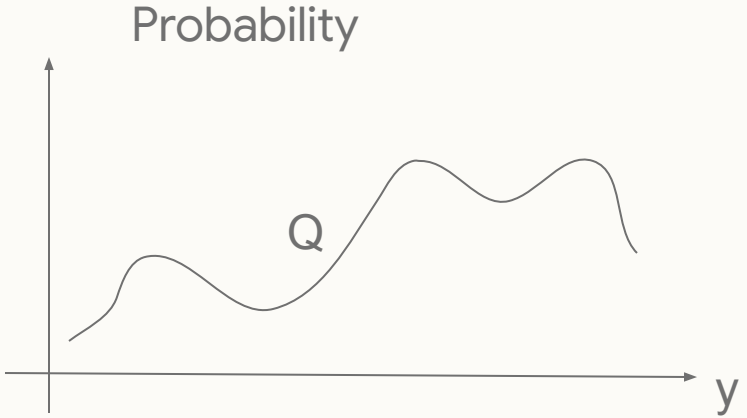
Comparing Distributions





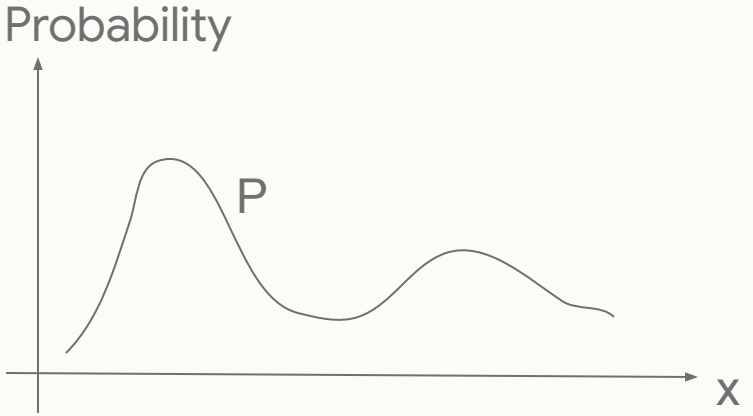
Comparing Distributions - Fréchet Distance

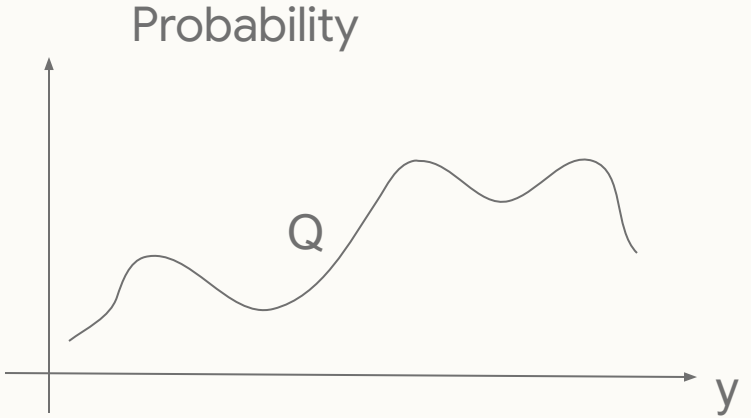




$$\text{dist}_F^2(P,Q) := \inf_{\gamma \in \Gamma(P,Q)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma} \|\mathbf{x}-\mathbf{y}\|^2$$

Comparing Distributions - Fréchet Distance





$$\mathrm{dist}_F^2(P,Q) := \inf_{\gamma \in \Gamma(P,Q)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma} \|\mathbf{x}-\mathbf{y}\|^2$$

$$\mathrm{dist}_F^2(P,Q) = \|\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q\|_2^2 + \mathrm{Tr}(\boldsymbol{\Sigma}_P + \boldsymbol{\Sigma}_Q - 2(\boldsymbol{\Sigma}_P\boldsymbol{\Sigma}_Q)^{\frac{1}{2}})$$

FID

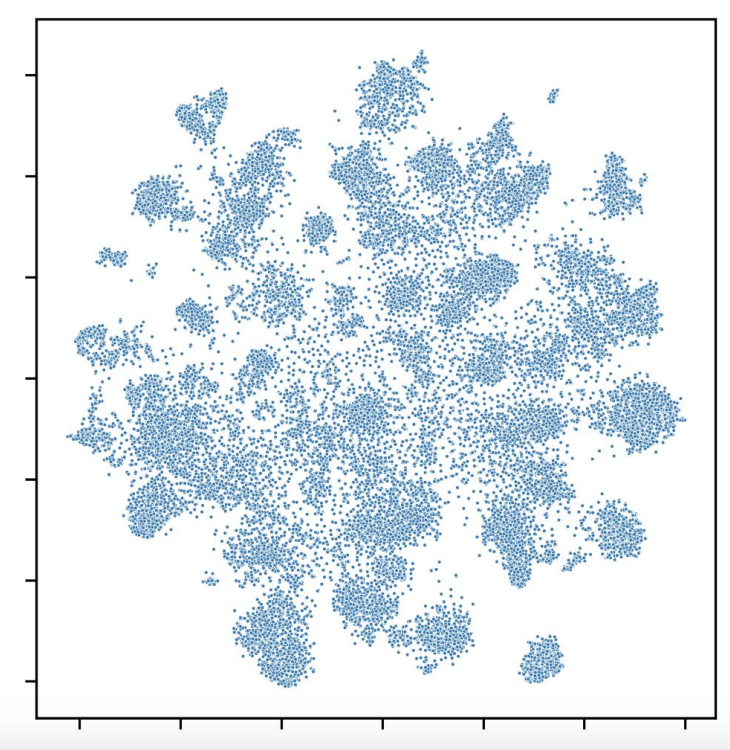
- Inception embeddings
  - Trained only on simple scenes from the Imagenet dataset
  - ~1M training images

- Fréchet distance
  - Gaussian assumption
  - Need to estimate a large (2048x2048) covariance matrix
  - Biased estimator [1]

---

[1] Min Jin Chong, David Forsyth. Effectively Unbiased FID and Inception Score and Where to Find Them, CVPR 2020.

Gaussian Assumption on Inception Embeddings



t-SNE visualization of Inception embeddings on the COCO 30K dataset

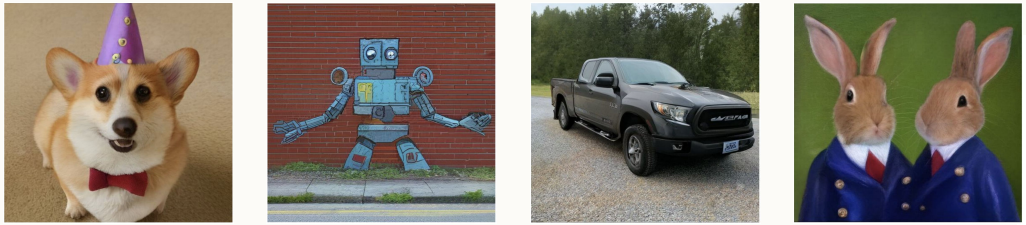
Test	Result
Mardia's Skewness Test	✗ Reject (p-value 0.0)
Mardia's Kurtosis Test	✗ Reject (p-value 0.0)
Henze-Zirkler Test	✗ Reject (p-value 0.0)

CMMD

	Fréchet distance	MMD distance
Inception Embeddings	✗ Weak image embeddings ✗ Incorrect normality assumption ✗ Sample inefficient ✗ Biased estimator **FID**	✗ Weak image embeddings ✓ Distribution-free ✓ Sample efficient ✓ Unbiased estimator
CLIP Embeddings	✓ Rich image embeddings ✗ Incorrect normality assumption ✗ Sample inefficient ✗ Biased estimator	✓ Rich image embeddings ✓ Distribution-free ✓ Sample efficient ✓ Unbiased estimator **CMMD**

Human Evaluation

Model-A



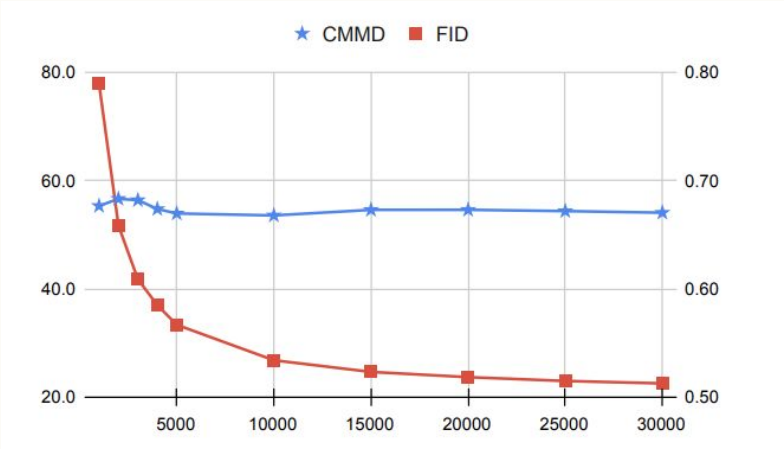
Model-B

Model	Model-A	Model-B
FID	21.40	18.42
FID_∞	20.16	17.19
KID	0.0105	0.0080
CMMD	0.721	0.951
Human rater preference	92.5%	6.9%

Table 3. *Human evaluation. FID and KID contradict human evaluation while CMMD agrees. Lower is better for all metrics.*

- CMMD correlates better with human perception of quality.

CMMD is More Efficient



Operation	Time
Fréchet distance	7007.59 ± 231 ms
MMD distance	71.42 ± 0.67 ms
Inception model inference	2.076 ± 0.15 ms
CLIP model inference	1.955 ± 0.14 ms

Table 4. *Comparing runtime for computing Fréchet/MMD distances and Inception/CLIP feature extractions.*

Conclusion

- Discrete token based image generations models integrate better with LLMs
- They can be made efficient using MRF-based structural prediction methods
- FID is far from ideal for image-generative model evaluation
- CMMD fixes some of FID's shortcomings

Thank you!