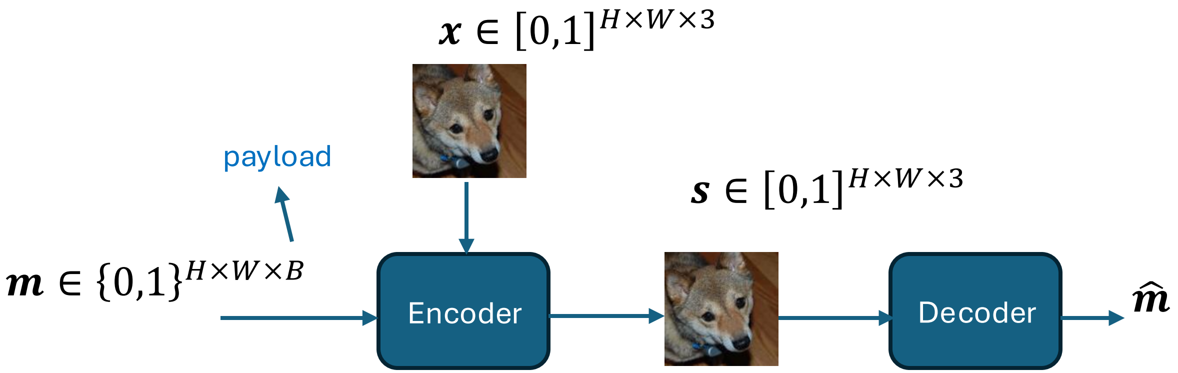
Neural Cover Selection for Image Steganography

Karl Chahine, Hyeji Kim

University of Texas at Austin

Background

- Image Steganography aims to hide a binary message $\boldsymbol{m}$ into a cover image $\boldsymbol{x}$ using an encoder, resulting in the steganographic image $\boldsymbol{s}$. The decoder estimates $\widehat{\boldsymbol{m}}$ from $\boldsymbol{s}$



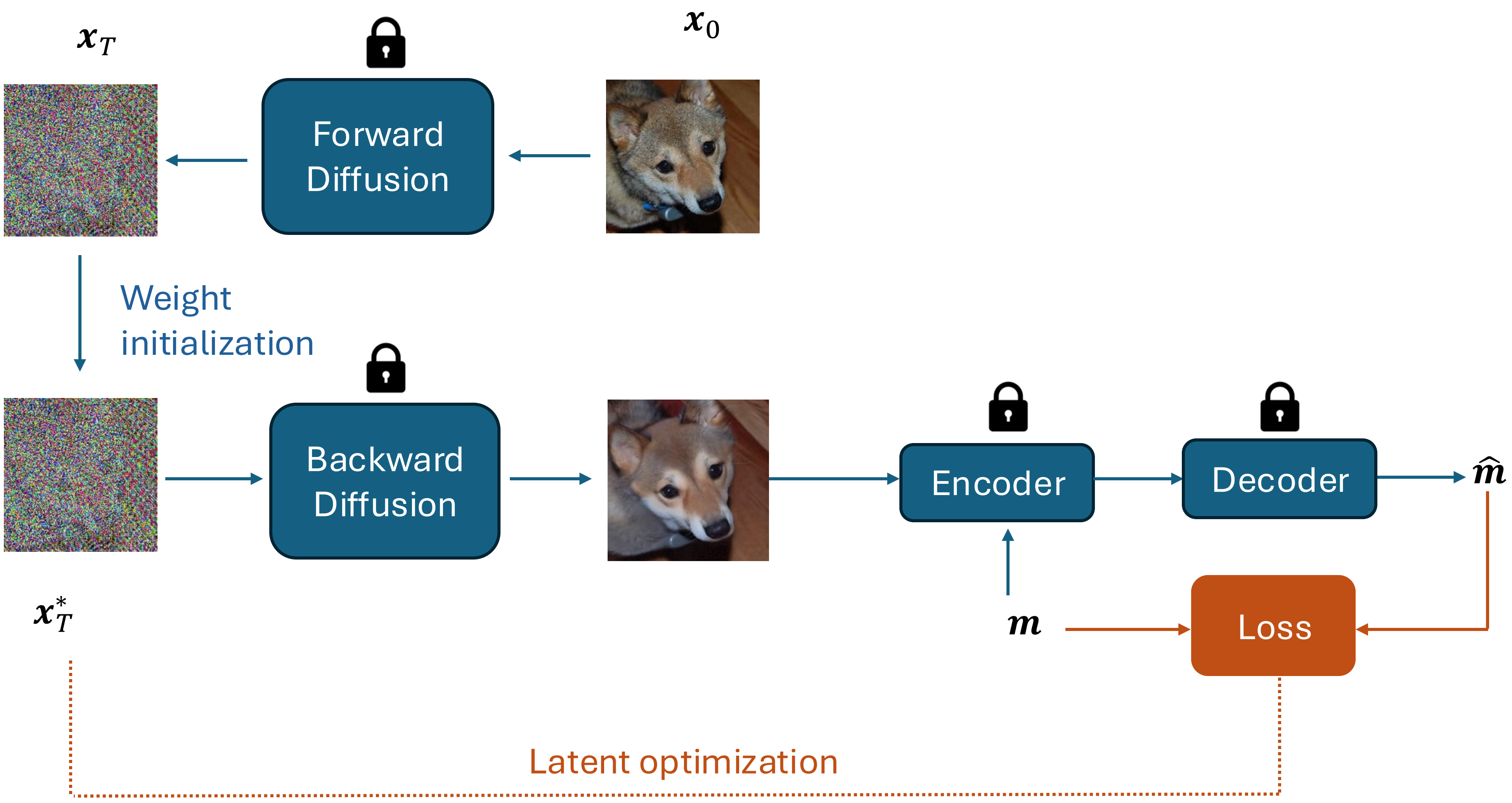
- The objectives are:
  - ➢ Design an encoder and decoder such that the error rate $\frac{||\boldsymbol{m}-\widehat{\boldsymbol{m}}||_0}{H\times W\times B}$ is minimized
  - ➢ $\boldsymbol{x}$ is visually identical to $\boldsymbol{s}$ to avoid steganalysis (the act of detecting if an image contains a secret message)

- Cover selection aims to find the best cover image $\boldsymbol{x}$ that achieves the above objectives for a given message $\boldsymbol{m}$
- In this work, we use a pretrained encoder-decoder pair

Prior Work

- Traditional algorithms empirically select cover images (based on DCT coefficients or image complexity measures)
- There is no clear optimal metric that correlates with good message hiding capabilities
- There is no understanding of how the steganographic encoder and decoder operate
- Contributions:
    1. Given a secret message, our algorithm optimizes over the latent space of a pretrained generative model, which in turns generates a suitable cover image
    2. We provide an analysis to investigate the encoder's hiding process

DDIM-based Cover Selection



Randomly Drawn Samples

Original

Error: 2.26%

Error: 0.42%

Error: 2.44%

Optimized

Error: 0.74%

Error: 0.11%

Error: 0.85%

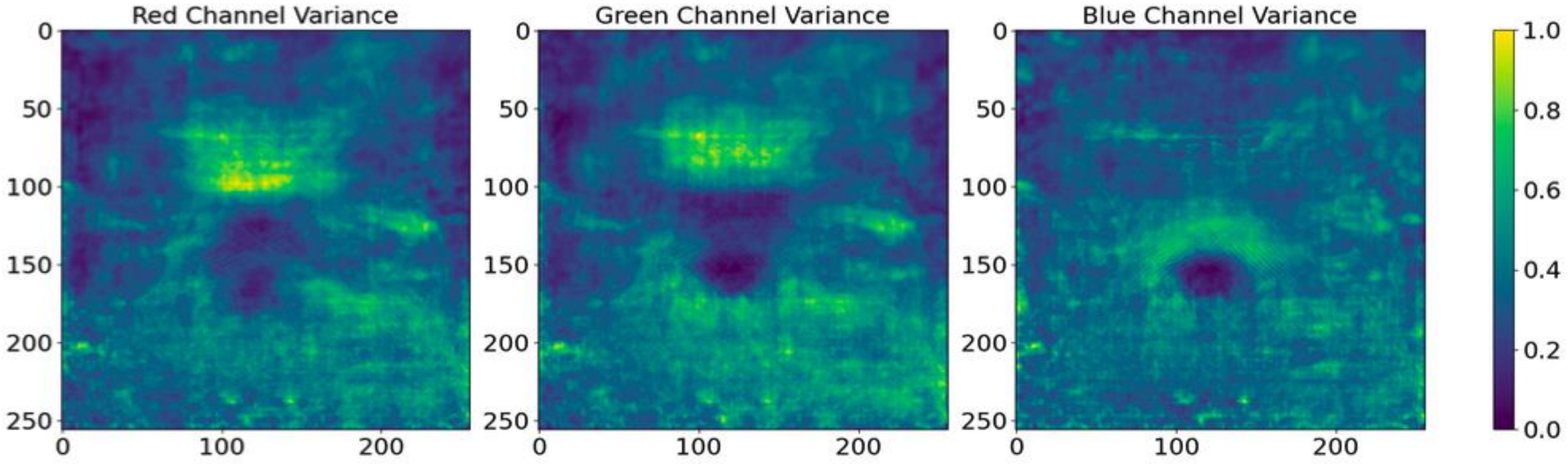
Analysis

- We hypothesize that the encoder preferentially hides messages in regions of low pixel variance. To test this, we use the ImageNet Robin class with a payload $B = 4$ bpp. We structure our analysis into 2 steps:

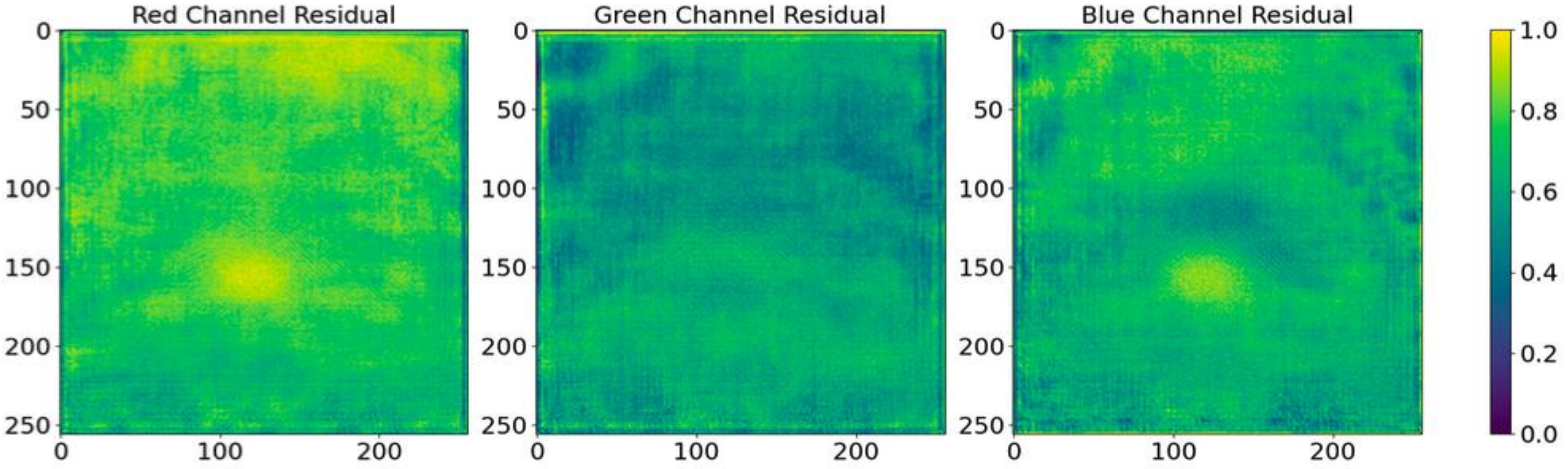
- Step 1: variance analysis. We calculate the variance of each pixel position for the 3 color channels across a batch of images and normalize to a range between 0 and 1

- Step 2: residual computation. Using the same batch of images, we pass them through the steganographic encoder to obtain the corresponding steganographic images. We then compute and normalize the absolute difference between them

Step 1:



Step 2:



- We quantize the features (variances < 0.5 are low variance pixels, residuals > 0.5 are high residual pixels)
- We find that 81.6% of high-residual pixels are encoded in low-variance pixels

Analogy to Waterfilling

- We draw parallels between our analysis and the waterfilling problem for Gaussian channels

- We consider a simple additive steganography scheme, where $m_i \in \{-1,1\}$ is the message to be embedded, γ_i represents its power, x_i and s_i represent the i-th element of the cover and steganographic images respectively:

$$s_i = x_i + \gamma_i m_i \quad i = 1,2,\ldots,N = H \times W \times 3$$

- We assume a power constraint P such that $E\left[\sum_{i=1}^{N}(s_i - x_i)^2\right] \leq P$

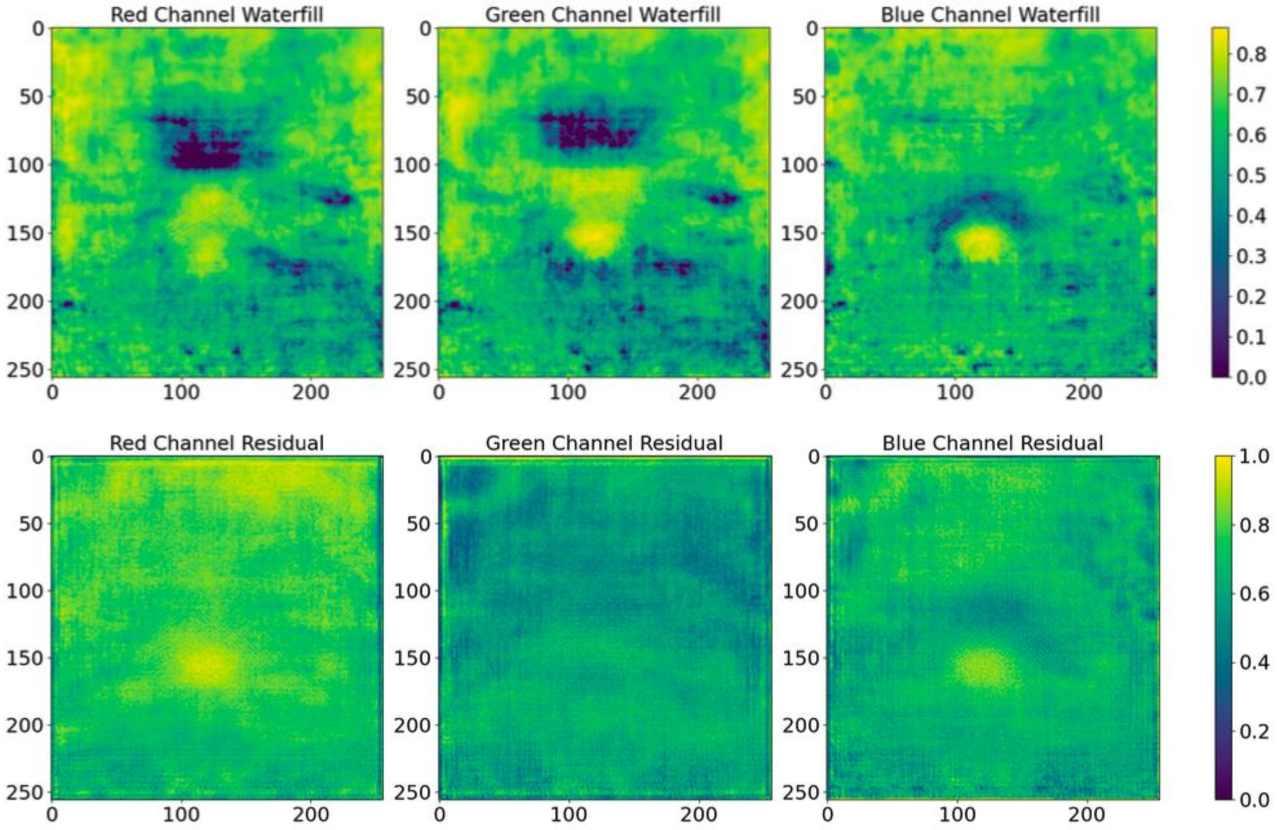
Analogy to Waterfilling

- Objective: distribute P among the N channels to maximize the capacity
  $C = \sum_{i=1}^{N} \log_2(1 + \frac{\gamma_i^2}{\sigma_i^2})$ where σ_i^2 is the variance of x_i

- Optimal allocation: $\gamma_i^2 = \left(\frac{1}{\lambda \ln(2)} \sigma_i^2\right)^+$, where $(x)^+ = \max(x, 0)$ and λ is chosen to satisfy the power constraint

- We calculate $\{\sigma_i^2\}_{i=1}^{H\times W\times 3}$ using a batch of images and find the optimized $\{\gamma_i^2\}_{i=1}^{H\times W\times 3}$ using the approach described above

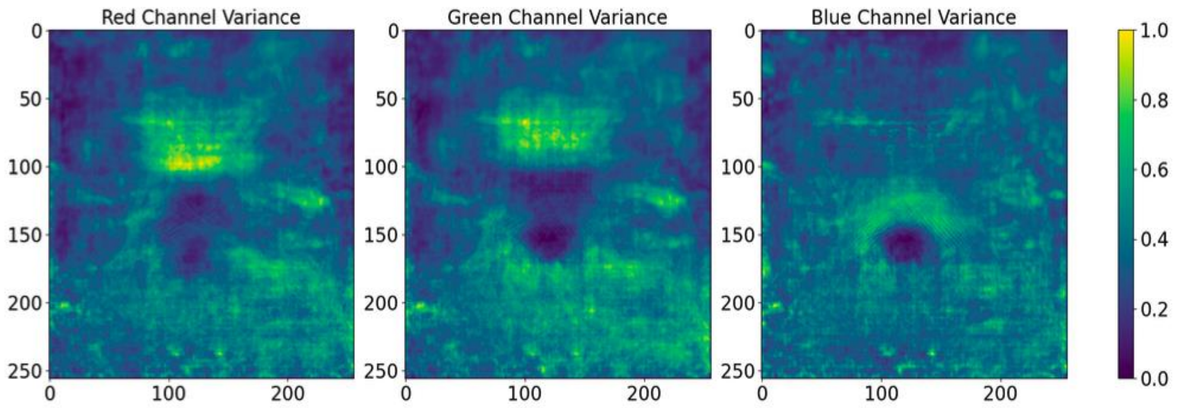
Analogy to Waterfilling



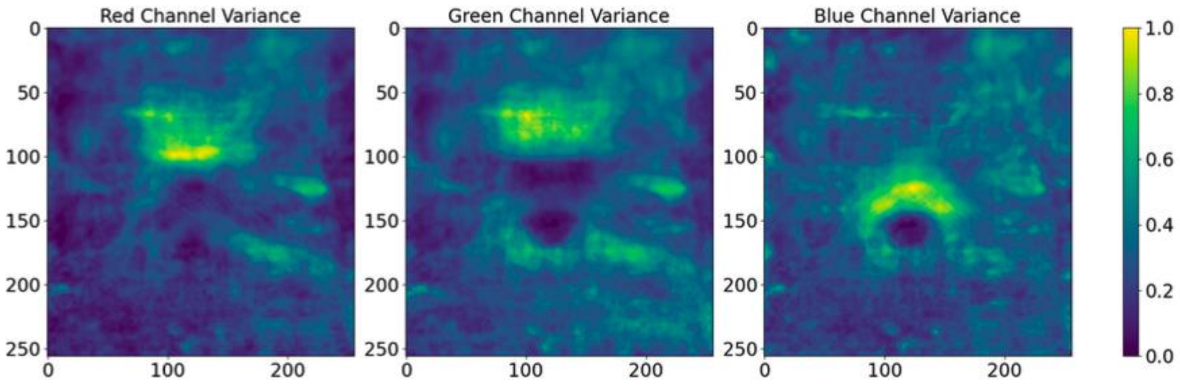
- We quantize the matrices by setting values greater than 0.5 to 1 and values less than 0.5 to 0
- Similarity scores across channels: **81.8%** (red), **65.5%** (green), **74.9%** (blue)

What's the Effect of Cover Optimization?

Before optimization



After optimization



- The number of low-variance spots significantly increased
- 92.4% of high-residual pixels are encoded in low-variance pixels as opposed to 81.6% before optimization

Thank you!