VALHALLA: Visual Hallucination for Machine Translation

Li et al. 2022

Presenter: Alvi Ishmam

CS 6804, 03/15/2023
Machine Translation

Translate to german: A bird is flying over the water
Ein Vogel fliegt über dem Wasser.

Multimodal Machine Translation

A bird flies over the water
Model
Ein Vogel fliegt über das Wasser

Use of computer systems that translates text or speech from one language to another

Input: text, image, audio..
Output: target language text
Machine Translation (MT) Systems

- Rule based ([Nyberg et al. 1992](#))
  - Knowledge based parsing and mapping form source to target language
- Statistical MT approaches ([Lopez et al. 2008](#), [Koehn 2007](#))
  - Mapping of source to target language done by confusion network - a decoder
- End to end neural networks ([Bandanau et al. 2014](#), [Sutskever et al. 2014](#))
  - Soft search of part of source sentence that are relevant to target sentence

However....

- These approaches only address text to text translation
- Other modalities provide useful rich semantic information
Overview

- RGB object detection model
- Multiple modalities produce better recognition model
- RGB image capturing devices are more pervasive than depth capturing devices
- The recognition model need to perform well on RGB images alone as input
Learning Side Information with Modality Hallucination

Contribution

- Incorporate additional side information (Depth information) as an image modality at training time
- A representation learning through hallucination network
- Final model only sees RGB images in test times
- Performs better than single modality in NYUD2 dataset
Related Works

- **RGB-D Detection**
  - Adding additional depth network representation into CNN ([Gupta et al. 2015](#))

- **Transfer Learning**
  - Mapping missing modalities through hallucination ([Christoudias et al. 2010](#))

- **Learning with Side Information**
  - Additional data/privileged information can be utilized in bonding box, image tags ([Sharmanska et al 2013](#))
Methodology: Overview

• Base model: Multilayer CNN for all networks
• Two modalities: 1) RGB 2) privileged depth modality with **Hallucination** network
• Hallucination net:
  • Inputs: RGB, image regions
  • Output: Detection scores for each category for each region

Figure 1: Training our modality hallucination architecture. We learn a multimodal Fast R-CNN [10] convolutional network for object detection. Our hallucination branch is trained to take an RGB input image and mimic the depth mid-level activations. The whole architecture is jointly trained with the bounding box labels and the standard softmax cross-entropy loss.

Figure 2: Test time modality hallucination architecture.
Methodology

Hallucination loss: \( \mathcal{L}_{\text{hallucinate}}(\ell) = \| \sigma(A_{\ell}^{\text{dNet}}) - \sigma(A_{\ell}^{\text{hNet}}) \|^2_{\text{2}} \)

Final Optimization

- 5 softmax cross entropy losses using bonding box labels as target, \( L_{\text{cls}} \)
- 5 smooth l1 loss using bonding box co-ordinates as targets, \( L_{\text{loc}} \)
- 1 additional hallucination loss

\[
\mathcal{L} = \gamma \mathcal{L}_{\text{hallucinate}} + \alpha \left[ \mathcal{L}_{\text{dNet}}^{\text{loc}} + \mathcal{L}_{\text{rNet}}^{\text{loc}} + \mathcal{L}_{\text{hNet}}^{\text{loc}} + \mathcal{L}_{\text{rdNet}}^{\text{loc}} + \mathcal{L}_{\text{hNet}}^{\text{loc}} \right] + \beta \left[ \mathcal{L}_{\text{cls}}^{\text{dNet}} + \mathcal{L}_{\text{cls}}^{\text{rNet}} + \mathcal{L}_{\text{cls}}^{\text{hNet}} + \mathcal{L}_{\text{cls}}^{\text{rdNet}} + \mathcal{L}_{\text{cls}}^{\text{hNet}} \right]
\]

Figure 1: Training our modality hallucination architecture. We learn a multimodal Fast R-CNN [10] convolutional network for object detection. Our hallucination branch is trained to take an RGB input image and mimic the depth mid-level activations. The whole architecture is jointly trained with the bounding box labels and the standard softmax cross-entropy loss.
Experiments

Datasets/ Base Network Setup

- RGB-D detection dataset NYUD2 (Ye 2013)
  - 1449 labeled image
- Base Network:
  - Fast R-CNN modified to AlexNet or VGG-1024 for all modules (RGB, depth, hallucination)
  - Depth images are encoded using HHA encoding
  - Depth network is fine-tuned after initializing with RGB weights
- Hyperparameters:
  - Update all layers with same learning rate
Experiments

NYUD2 Detection Evaluation

<table>
<thead>
<tr>
<th>method</th>
<th>btub</th>
<th>bed</th>
<th>bshelf</th>
<th>box</th>
<th>chair</th>
<th>counter</th>
<th>desk</th>
<th>door</th>
<th>dresser</th>
<th>gbin</th>
<th>lamp</th>
<th>monitor</th>
<th>nstand</th>
<th>pillow</th>
<th>sink</th>
<th>sofa</th>
<th>table</th>
<th>tv</th>
<th>toilet</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB only [10] (A)</td>
<td>7.5</td>
<td>50.6</td>
<td>36.8</td>
<td>1.4</td>
<td>30.2</td>
<td>34.9</td>
<td>10.8</td>
<td>21.5</td>
<td>27.8</td>
<td>16.9</td>
<td>26.0</td>
<td>32.6</td>
<td>20.6</td>
<td>25.1</td>
<td>31.6</td>
<td>36.7</td>
<td>14.8</td>
<td>25.1</td>
<td>54.6</td>
<td>26.6</td>
</tr>
<tr>
<td>RGB ensemble (A-A)</td>
<td>10.5</td>
<td>53.7</td>
<td>33.6</td>
<td>1.6</td>
<td>32.0</td>
<td>34.8</td>
<td>12.2</td>
<td>20.8</td>
<td>34.5</td>
<td>19.6</td>
<td>28.6</td>
<td>45.7</td>
<td>28.5</td>
<td>24.4</td>
<td>31.4</td>
<td>34.7</td>
<td>14.5</td>
<td>34.0</td>
<td>56.1</td>
<td>29.0</td>
</tr>
<tr>
<td>Our Net (A-RGB, A-H)</td>
<td><strong>13.9</strong></td>
<td><strong>56.1</strong></td>
<td><strong>34.4</strong></td>
<td><strong>1.9</strong></td>
<td><strong>32.9</strong></td>
<td><strong>40.5</strong></td>
<td><strong>12.9</strong></td>
<td><strong>22.6</strong></td>
<td><strong>37.4</strong></td>
<td><strong>22.0</strong></td>
<td><strong>28.9</strong></td>
<td><strong>46.2</strong></td>
<td><strong>31.9</strong></td>
<td><strong>22.9</strong></td>
<td><strong>34.2</strong></td>
<td><strong>34.2</strong></td>
<td><strong>19.4</strong></td>
<td><strong>33.2</strong></td>
<td><strong>53.6</strong></td>
<td><strong>30.5</strong></td>
</tr>
<tr>
<td>RGB only [10] (V)</td>
<td>15.6</td>
<td>59.4</td>
<td>38.2</td>
<td>1.9</td>
<td>33.8</td>
<td>36.3</td>
<td>12.1</td>
<td>24.5</td>
<td>31.6</td>
<td>18.6</td>
<td>25.5</td>
<td>46.5</td>
<td>30.1</td>
<td>20.6</td>
<td>30.3</td>
<td>40.5</td>
<td>19.5</td>
<td>37.8</td>
<td>45.7</td>
<td>29.9</td>
</tr>
<tr>
<td>RGB ensemble (A-V)</td>
<td>14.8</td>
<td>60.4</td>
<td><strong>43.1</strong></td>
<td><strong>2.1</strong></td>
<td>36.4</td>
<td>40.7</td>
<td>13.3</td>
<td><strong>27.1</strong></td>
<td>35.5</td>
<td>20.8</td>
<td>29.9</td>
<td><strong>52.9</strong></td>
<td><strong>33.5</strong></td>
<td>26.2</td>
<td>33.0</td>
<td>44.4</td>
<td>19.9</td>
<td>36.7</td>
<td>50.2</td>
<td>32.7</td>
</tr>
<tr>
<td>Our Net (A-RGB, V-H)</td>
<td><strong>16.8</strong></td>
<td><strong>62.3</strong></td>
<td><strong>41.8</strong></td>
<td><strong>2.1</strong></td>
<td><strong>37.3</strong></td>
<td><strong>43.4</strong></td>
<td><strong>15.4</strong></td>
<td><strong>24.4</strong></td>
<td><strong>39.1</strong></td>
<td><strong>22.4</strong></td>
<td><strong>30.3</strong></td>
<td><strong>46.6</strong></td>
<td><strong>30.9</strong></td>
<td><strong>27.0</strong></td>
<td><strong>42.9</strong></td>
<td><strong>46.2</strong></td>
<td><strong>22.2</strong></td>
<td><strong>34.1</strong></td>
<td><strong>60.4</strong></td>
<td><strong>34.0</strong></td>
</tr>
</tbody>
</table>

Table 1: **Detection (AP%) on NYUD2 test set:** We compare our performance (pool5 hallucinate) against a Fast R-CNN [10] RGB detector trained on NYUD2 and against an ensemble of Fast R-CNN RGB detectors. AlexNet [21] architecture is denoted as ‘A’ and VGG-1024 [10, 28] architecture is denoted as ‘V’. Our method outperforms both the RGB-only baselines and the RGB ensemble baselines.
## Experiments

### Hallucination net initialization

<table>
<thead>
<tr>
<th>Initial Weights</th>
<th>bathtub</th>
<th>bed</th>
<th>bshelf</th>
<th>box</th>
<th>chair</th>
<th>counter</th>
<th>desk</th>
<th>door</th>
<th>dresser</th>
<th>gbin</th>
<th>lamp</th>
<th>monitor</th>
<th>nstand</th>
<th>pillow</th>
<th>sink</th>
<th>sofa</th>
<th>table</th>
<th>tv</th>
<th>toilet</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>7.5</td>
<td>50.4</td>
<td>9.9</td>
<td>0.9</td>
<td>26.2</td>
<td>24.9</td>
<td>5.8</td>
<td>15.8</td>
<td>13.0</td>
<td>29.8</td>
<td>12.0</td>
<td>43.1</td>
<td>20.9</td>
<td>14.7</td>
<td>17.9</td>
<td>25.3</td>
<td>15.1</td>
<td>32.5</td>
<td>59.1</td>
<td>22.4</td>
</tr>
<tr>
<td>depth</td>
<td>9.9</td>
<td>52.4</td>
<td>14.9</td>
<td>0.9</td>
<td>24.9</td>
<td>24.4</td>
<td>4.3</td>
<td>15.3</td>
<td>18.1</td>
<td>24.1</td>
<td>14.8</td>
<td>45.8</td>
<td>27.2</td>
<td>18.5</td>
<td>21.3</td>
<td>29.0</td>
<td>13.7</td>
<td>33.6</td>
<td>66.4</td>
<td>24.2</td>
</tr>
<tr>
<td>random</td>
<td>10.5</td>
<td>47.6</td>
<td>12.3</td>
<td>0.6</td>
<td>23.5</td>
<td>20.2</td>
<td>6.0</td>
<td>13.0</td>
<td>19.3</td>
<td>12.0</td>
<td>13.3</td>
<td>42.8</td>
<td>12.8</td>
<td>12.1</td>
<td>13.6</td>
<td>23.0</td>
<td>13.9</td>
<td>28.6</td>
<td>61.5</td>
<td>20.3</td>
</tr>
</tbody>
</table>

Table 2: **RGB Detection (AP%) on NYUD2 val set**: We compare initializing the hallucination network by randomly initializing or by using the pre-trained RGB or depth parameter values.
## Experiments

### Which layer to Hallucinate

<table>
<thead>
<tr>
<th>hallucination layer</th>
<th>bathtub</th>
<th>bed</th>
<th>bshelf</th>
<th>box</th>
<th>chair</th>
<th>counter</th>
<th>desk</th>
<th>door</th>
<th>dresser</th>
<th>gbin</th>
<th>lamp</th>
<th>monitor</th>
<th>nstand</th>
<th>pillow</th>
<th>sink</th>
<th>sofa</th>
<th>table</th>
<th>tv</th>
<th>toilet</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB only</td>
<td>4.9</td>
<td>45.5</td>
<td>10.9</td>
<td>1.3</td>
<td>21.5</td>
<td>23.6</td>
<td>5.4</td>
<td>14.5</td>
<td>12.7</td>
<td>17.4</td>
<td>9.4</td>
<td>40.9</td>
<td>17.2</td>
<td>14.9</td>
<td>19.9</td>
<td>19.2</td>
<td>14.0</td>
<td>32.5</td>
<td>66.3</td>
<td>20.6</td>
</tr>
<tr>
<td>pool1</td>
<td>12.0</td>
<td>54.0</td>
<td>17.9</td>
<td>1.1</td>
<td>24.5</td>
<td>23.6</td>
<td>5.0</td>
<td>15.2</td>
<td>16.3</td>
<td>12.7</td>
<td>13.3</td>
<td>40.0</td>
<td>24.7</td>
<td>16.6</td>
<td>20.5</td>
<td>29.6</td>
<td>14.9</td>
<td>27.4</td>
<td>55.3</td>
<td>22.3</td>
</tr>
<tr>
<td>pool2</td>
<td>8.4</td>
<td>50.7</td>
<td>13.5</td>
<td>1.0</td>
<td>24.2</td>
<td>26.0</td>
<td>6.6</td>
<td>13.1</td>
<td>13.8</td>
<td>17.8</td>
<td>11.7</td>
<td>40.7</td>
<td>21.8</td>
<td>15.0</td>
<td>20.5</td>
<td>22.4</td>
<td>15.2</td>
<td>27.2</td>
<td>59.7</td>
<td>21.5</td>
</tr>
<tr>
<td>conv3</td>
<td>8.8</td>
<td>52.5</td>
<td>13.2</td>
<td>1.0</td>
<td>25.6</td>
<td>26.2</td>
<td>3.3</td>
<td>13.2</td>
<td>14.9</td>
<td>17.0</td>
<td>16.2</td>
<td>41.6</td>
<td>22.2</td>
<td>20.2</td>
<td>22.9</td>
<td>24.6</td>
<td>17.2</td>
<td>37.4</td>
<td>65.6</td>
<td>23.3</td>
</tr>
<tr>
<td>conv4</td>
<td>9.7</td>
<td>51.2</td>
<td>12.9</td>
<td>1.0</td>
<td>26.3</td>
<td>26.8</td>
<td>6.9</td>
<td>17.4</td>
<td>16.7</td>
<td>22.0</td>
<td>12.4</td>
<td>43.2</td>
<td>15.5</td>
<td>16.4</td>
<td>24.0</td>
<td>23.5</td>
<td>16.2</td>
<td>34.2</td>
<td>64.2</td>
<td>23.2</td>
</tr>
<tr>
<td>pool5</td>
<td>9.9</td>
<td>52.4</td>
<td>14.9</td>
<td>0.9</td>
<td>24.9</td>
<td>24.4</td>
<td>4.3</td>
<td>15.3</td>
<td>18.1</td>
<td>24.1</td>
<td>14.8</td>
<td>45.8</td>
<td>27.2</td>
<td>18.5</td>
<td>21.3</td>
<td>29.0</td>
<td>13.7</td>
<td>33.6</td>
<td>66.4</td>
<td>24.2</td>
</tr>
<tr>
<td>fc6</td>
<td>10.3</td>
<td>47.2</td>
<td>12.0</td>
<td>0.6</td>
<td>21.7</td>
<td>20.0</td>
<td>5.9</td>
<td>12.8</td>
<td>13.8</td>
<td>20.5</td>
<td>11.8</td>
<td>34.4</td>
<td>16.3</td>
<td>13.1</td>
<td>14.8</td>
<td>27.3</td>
<td>16.1</td>
<td>28.8</td>
<td>60.5</td>
<td>20.4</td>
</tr>
<tr>
<td>fc7</td>
<td>3.3</td>
<td>49.4</td>
<td>12.7</td>
<td>0.8</td>
<td>24.1</td>
<td>21.8</td>
<td>4.8</td>
<td>15.2</td>
<td>16.8</td>
<td>11.7</td>
<td>10.0</td>
<td>43.4</td>
<td>18.7</td>
<td>14.2</td>
<td>20.6</td>
<td>25.2</td>
<td>14.4</td>
<td>29.5</td>
<td>63.1</td>
<td>21.0</td>
</tr>
<tr>
<td>fc8</td>
<td>4.2</td>
<td>50.7</td>
<td>13.9</td>
<td>0.9</td>
<td>23.8</td>
<td>23.6</td>
<td>5.4</td>
<td>15.5</td>
<td>18.0</td>
<td>13.2</td>
<td>13.3</td>
<td>42.0</td>
<td>20.9</td>
<td>15.8</td>
<td>22.3</td>
<td>23.8</td>
<td>14.5</td>
<td>29.6</td>
<td>63.6</td>
<td>21.8</td>
</tr>
</tbody>
</table>

**Table 3: RGB Detection (AP%) on NYUD2 val set:** We compare hallucinating different mid-level features with our method.
Hallucination networks improves object detection performance by adding additional side information to the model.
Motivation

• Machine Translation system relies on text only
• Lack explicit grounding from other modalities
• Additional visual context should help MT
• Grounding should improve data efficiency
• Benefit translation process in low resource settings
Related Works

• Multimodal Machine Translation
 ✦ Visual alignment for unsupervised word mapping ([Sigurdsson et al. 2020](#))
 ✦ Beneficial under limited textual context ([Caglayan 2019, Elliot et al. 2018](#))

• Vision-Language Learning
 ✦ Visual grounding is used for MT in many tasks e.g., semantic parsing, coreference resolution, representation learning ([Shi et al. 2019](#))
 ✦ Still improving MT with no visual information in test time remains as an open challenge

• Modality Hallucination
  • Similar concept is applied in object detection task where depth features are hallucinated from RGB input ([Hoffman 2016](#))
Methodology

Traditional Machine Translation (Text to Text)

• Input source sentence $x = (x_1, \ldots, x_S)$,
• Output target sentence $y = (y_1, \ldots, y_T)$
• Transformer, $f_T = (f_T^{enc}, f_T^{dec})$
• The likelihood of target tokens conditioned on input sentence
• Transformer is trained optimizing cross entropy loss

$$p(y \mid x; f_T) = \prod_{i=1}^{T} f_T(y_i \mid y_{<i}, x) \quad \Delta \quad \prod_{i=1}^{T} f_T^{dec}(y_i \mid y_{<i}, f_T^{enc}(x)),$$
Methodology

Multimodal Machine Translation (Text + Visual to Text)

$p(y \mid x; f_T) = \prod_{i=1}^{T} f_T(y_i \mid y_{<i}, x)$

$\Delta = \prod_{i=1}^{T} f_{T \text{dec}}^i (y_i \mid y_{<i}, f_T^{\text{enc}}(x)),$

Text to Text

Text + Visual to Text

$\begin{align*}
\text{Text to Text} \\
\downarrow \\
\text{Text + Visual to Text}
\end{align*}$

- Input source sentence $x = (x_1, \ldots, x_S)$,
- Input visual information: $v$
- Output target sentence $y = (y_1, \ldots, y_T)$
- Transformer, $f_T = (f_T^{\text{enc}}, f_T^{\text{dec}})$
- Use encoder, $f_v$ to map into latent visual representation $z = f_v(v)$
- Transformer is trained optimizing cross entropy loss
- **Issues**: requires both text and image during inference
Methodology: VALHALLA Architecture

Figure 2: Overview of VALHALLA Architecture for Machine Translation. **Left:** Training pipeline of VALHALLA. Translation outputs are gathered from two streams of input, either with ground-truth visual tokens $z$ or hallucinated representation $\hat{z}$, and optimized on a combination of hallucination, translation and consistency losses. **Right:** Inference process of VALHALLA in the absence of visual inputs.
Methodology: VALLHALLA Architecture

Choice of Visual Encoder: Continuous vs Discrete

• Traditionally combine text input with *continuous* visual embeddings extracted from pre-trained models e.g., ResNet50, ViT

• **Issues:**
  ✫ Difficult to concatenate with discrete word embeddings with continuous visual embeddings
  ✫ Complex aggregation module is required
  ✫ Training instability
  ✫ Model prediction collapse to mean value

Idea:

Combine discrete text embeddings with *discrete* visual encoder

• Easy to aggregate between the modalities
• Easier to optimize with vanilla cross entropy loss
Methodology: VALLHALLA Architecture

Discrete Visual Encoder: Autoencoder

- Due to overfitting, the latent space can be very irregular
- Close points in latent space can give different decoded data
- Is not reliable for generating new data

Distribution over the latent space instead a single point
- Add a regularization term to loss function for better organization

Image source: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510910f73
Methodology: VALLHALLA Architecture

Variational Quantization GAN with VAE: VQGAN VAE (ESSER et al. 2021)

- Learn to represent image with codes from a learned, discrete codebook, \( Z \)
- A codebook provides an interface between transformers
- Obtain the representation from the encoder and subsequent element-wise quantization onto its closest codebook entry

- High resolution image requires globally consistent and locally realistic pattern generation from the model
- Pixel level representation is not sufficient
- Representation of rich semantic information in a form of codebook
- Codebook constituents example: Bounding box of mouth, nose, ear, rather than a full face
Methodology: VALLHALLA Architecture

Discrete Visual Encoder based on VQGAN VAE

• Implement vector quantization with VQGAN VAE

\[ z = Q(f_V(v); E_V). \]

• Q is quantization function and Ev is d dimensional visual codebook of size k

\[ Q_i(c; E_V) = \arg \min_{k \in \{1, \ldots, K\}} \| c_i - e^{(k)} \|_2 \]

• Maps each spatial location of feature array c to the closest index of codebook e
Methodology: VALLHALLA Architecture

Visual Hallucination:

• VALHALLA relies on this module when no visual information is available during testing
• Uses an autoregressive transformer (Ramesh et al. 2021)
• Autoregressively model text and image tokens as a single stream of data
• Maximize the joint likelihood of $x$ and $z$ by optimizing the cross-entropy hallucination loss
• Final hallucinated visual sequences is defined by the most likely tokens predicted by $f_h$ at each timestamp

\[
\ell_H(f_H) = \mathbb{E}_{(x,z)} [-\log p(x, z; f_H)].
\]

\[
p(x, z; f_H) = p(x; f_H)p(z | x; f_H)
= \prod_{i=1}^{S} f_H(x_i | x_{<i}) \prod_{j=1}^{V} f_H(z_j | z_{<j}, x).
\]

\[
\hat{z}_i = \arg\max_{k \in \{1,...,K\}} f_H(z_i = k | z_{<i}, x),
\]
Methodology: VALLHALLA Architecture

Visual Hallucination cont

• **Issue:** Creates mismatch between training and testing when decoding is performed only on source text tokens

• **Constancy loss is computed:**

\[
\ell_C(f_H, f_T) = \mathbb{E}_{(x, z, y)} \left[ \sum_{i=1}^{T} \text{KL}[y_i^M \parallel y_i^H] \right]
\]

• \(y_i^M, y_i^H\) are the next word distributions from ground-truth visual tokens and hallucinated features respectively,

• \(\text{KL}[y_i^M \parallel y_i^H]\) is the Kullback-Leibler divergence between the two conditional distributions.
Methodology: VALLHALLA Architecture

Visual Hallucination cont: Optimization

• Joint optimization of consistency loss and translation loss is critical
• The output of the hallucination transformer ($\hat{z}$) prevents loss gradients from back propagating.
• Idea: Relax the training objective rather than taking the most likely predicted tokens from the transformer
• Gumbel-softmax relaxation (Maddison 2016) during training,

$$\hat{z}_i = \sum_{k=1}^{K} \frac{\exp((\log \pi_{i,k} + g_k)/\tau)}{\sum_i \exp((\log \pi_{i,i} + g_i)/\tau)} o_k.$$  

Final optimization objective of VALHALLA:

$$\ell(f_H, f_T) = \ell_T(f_T; z) + \ell_T(f_T; \hat{z}) + \gamma_H \ell_H(f_H) + \lambda_C \ell_C(f_H, f_T),$$

$\tau$ is the temperature of the softmax

$o_k$ is the hot vector of length $k$, activated at dimension $k$, $g_1,g_2, ... ~ \text{Gumbel (0,1)}$
Experiments

Datasets and Tasks

- **Datasets** - Multi30K ([Elliot 2016](#)), Wikipedia Image Text (WIT) ([Srinivasan et al. 2021](#)) and WMT2014 ([Bojar 2014](#))
- Three test split; **Test2016, Test2017, MSCOCO**
- WIT is created by extracting text-image pairs from Wikipedia
- New benchmarks using three different settings
- **Tasks**: Well resourced (EN -> {DE, FR}), Under resourced ({EN -> RO}), Non English (DE-> ES)
- WMT does not provide aligned image
- Used CLIP ([Radford et al. 2021](#)) to retrieve top-5 images from Multi30K or WIT datasets to train transformers.
Experiments

Models, Baselines

- **Models**: Three variants; Base, Small, Tiny
- Multi30K: Small and Tiny due to better performance
- WMT/ WIT: Base (Well resourced), small (under/non English task)

- **Baselines**:
  - Text Only: train transformer without visual information
  - Conventional MMT models (e.g., DCCN ([Lin et al. 2020](#)), GMNMT ([Yin et al. 2020](#)), and Gated Fusion ([Wu et al. 2021](#))
  - Text only inputs during test time ImagIT ([Long et al. 2021](#)), UVR-NMT ([Fang et al. 2022](#))

- Evaluation Metrics:
  - BLEU - n-gram matching with ground truth
  - METOR - considers semantic similarity between generated text and ground truth
## Results

### Multi30k

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>EN $\rightarrow$ DE</th>
<th></th>
<th>EN $\rightarrow$ FR</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Test2016</td>
<td>Test2017</td>
<td>MSCOCO</td>
<td>Average</td>
</tr>
<tr>
<td>Transformer-Base</td>
<td>T</td>
<td>32.0 ± 0.9</td>
<td>23.3 ± 0.8</td>
<td>21.3 ± 0.9</td>
<td>25.5 ± 0.9</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>39.4 ± 0.3</td>
<td>31.7 ± 0.2</td>
<td>27.9 ± 0.3</td>
<td>30.9 ± 0.4</td>
</tr>
<tr>
<td></td>
<td>VM</td>
<td>39.6 ± 0.3</td>
<td>31.8 ± 0.2</td>
<td>27.9 ± 0.3</td>
<td>33.1 ± 0.3</td>
</tr>
<tr>
<td>Transformer-Small</td>
<td>T</td>
<td>39.7 ± 0.3</td>
<td>31.7 ± 0.5</td>
<td>28.4 ± 0.2</td>
<td>33.3 ± 0.3</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>41.9 ± 0.2</td>
<td>34.0 ± 0.3</td>
<td>30.3 ± 0.3</td>
<td>35.4 ± 0.3</td>
</tr>
<tr>
<td></td>
<td>VM</td>
<td>41.9 ± 0.2</td>
<td>34.0 ± 0.3</td>
<td>30.4 ± 0.4</td>
<td>35.4 ± 0.3</td>
</tr>
</tbody>
</table>

Table 1: **BLEU score on Multi30K.** T: Baseline text-only transformer; V: **VALHALLA** model with hallucinated visual representations; VM: **VALHALLA** model with ground-truth visual representations. Please refer to supplementary material for METEOR score comparisons.
## Results

**Multi30k cont:**

<table>
<thead>
<tr>
<th>Method</th>
<th>EN $\rightarrow$ DE</th>
<th></th>
<th>EN $\rightarrow$ FR</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test2016</td>
<td>BLEU</td>
<td>METEOR</td>
<td>Test2017</td>
</tr>
<tr>
<td>Gumbel-Attention [31]</td>
<td>39.2</td>
<td>57.8</td>
<td>31.4</td>
<td>51.2</td>
</tr>
<tr>
<td>CAP-ALL [29]</td>
<td>39.6</td>
<td>57.5</td>
<td>33.0</td>
<td>52.2</td>
</tr>
<tr>
<td>GMNMT [69]</td>
<td>39.8</td>
<td>57.6</td>
<td>32.2</td>
<td>51.9</td>
</tr>
<tr>
<td>DCCN [30]</td>
<td>39.7</td>
<td>56.8</td>
<td>31.0</td>
<td>49.9</td>
</tr>
<tr>
<td><strong>VALHALLA (M)</strong></td>
<td><strong>41.9</strong></td>
<td><strong>68.7</strong></td>
<td><strong>34.0</strong></td>
<td><strong>62.5</strong></td>
</tr>
<tr>
<td>Gated Fusion [66]</td>
<td>42.0</td>
<td>67.8</td>
<td>33.6</td>
<td>61.9</td>
</tr>
<tr>
<td><strong>VALHALLA (M)</strong></td>
<td><strong>42.6</strong></td>
<td><strong>69.3</strong></td>
<td><strong>35.1</strong></td>
<td><strong>62.8</strong></td>
</tr>
</tbody>
</table>

**Text-Only Machine Translation**

<table>
<thead>
<tr>
<th>Method</th>
<th>EN $\rightarrow$ DE</th>
<th></th>
<th>EN $\rightarrow$ FR</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test2016</td>
<td>BLEU</td>
<td>METEOR</td>
<td>Test2017</td>
</tr>
<tr>
<td>VMMT$_r$ [7]</td>
<td>37.7</td>
<td>56.0</td>
<td>30.1</td>
<td>49.9</td>
</tr>
<tr>
<td>UAV-NMT [74]</td>
<td>36.9</td>
<td>–</td>
<td>28.6</td>
<td>–</td>
</tr>
<tr>
<td>ImagiT [32]</td>
<td>38.5</td>
<td>55.7</td>
<td>32.1</td>
<td>52.4</td>
</tr>
<tr>
<td><strong>VALHALLA</strong></td>
<td><strong>41.9</strong></td>
<td><strong>68.8</strong></td>
<td><strong>34.0</strong></td>
<td><strong>62.5</strong></td>
</tr>
<tr>
<td>RMMT [66]</td>
<td>41.4</td>
<td>68.0</td>
<td>32.9</td>
<td>61.7</td>
</tr>
<tr>
<td><strong>VALHALLA</strong></td>
<td><strong>42.7</strong></td>
<td><strong>69.3</strong></td>
<td><strong>35.1</strong></td>
<td><strong>62.8</strong></td>
</tr>
</tbody>
</table>

Table 2: Comparison with state-of-the-art multimodal and text-only translation methods on Multi30K. **VALHALLA** hallucinates visual representations from text-only inputs, while **VALHALLA (M)** uses ground-truth visual tokens at test time. Results in gray are computed with model averaging over 10 latest checkpoints. **VALHALLA** establishes new state-of-the-art for machine translation on Multi30K.
Results

Multi30k cont:

![Graphs showing BLEU scores for EN-DE and EN-FR with different sentence lengths.]

Figure 3: BLEU scores on different groups divided according to source sentence lengths on Multi30K Test2016 split.
## Results

### WIT/WMT

<table>
<thead>
<tr>
<th>Method</th>
<th>Well-Resourced</th>
<th>Non-English</th>
<th>Under-Resourced</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EN $\rightarrow$ DE</td>
<td>EN $\rightarrow$ ES</td>
<td>EN $\rightarrow$ FR</td>
<td></td>
</tr>
<tr>
<td>Text-Only</td>
<td>16.0 ± 0.5</td>
<td>24.8 ± 0.8</td>
<td>16.1 ± 1.2</td>
<td></td>
</tr>
<tr>
<td>UVR-NMT [74]</td>
<td>16.9 ± 0.2</td>
<td>26.4 ± 0.4</td>
<td>17.7 ± 0.3</td>
<td>15.1 ± 0.6</td>
</tr>
<tr>
<td>RMMT [66]</td>
<td>16.4 ± 0.3</td>
<td>24.8 ± 0.4</td>
<td>17.2 ± 1.6</td>
<td>16.1 ± 0.7</td>
</tr>
<tr>
<td>VALHALLA</td>
<td>17.5 ± 0.4</td>
<td>27.5 ± 0.2</td>
<td>18.8 ± 0.2</td>
<td>17.2 ± 0.4</td>
</tr>
<tr>
<td>VALHALLA (M)</td>
<td>17.4 ± 0.4</td>
<td>27.5 ± 0.2</td>
<td>18.8 ± 0.2</td>
<td>17.2 ± 0.4</td>
</tr>
</tbody>
</table>

|                  | EN $\rightarrow$ ES    | ES $\rightarrow$ FR | EN $\rightarrow$ RO | EN $\rightarrow$ AF |
| Text-Only        | 10.7 ± 0.2              | 16.2 ± 0.3           | 11.5 ± 0.7            | 10.8 ± 0.6   |
| UVR-NMT [74]     | 10.9 ± 0.9              | 16.4 ± 0.6           | 12.5 ± 0.5            | 11.6 ± 1.7   |
| RMMT [66]        | 11.0 ± 0.3              | 15.9 ± 0.7           | 9.9 ± 1.4             | 9.8 ± 1.0    |
| VALHALLA         | 11.3 ± 0.2              | 16.6 ± 0.8           | 14.4 ± 1.0            | 14.0 ± 0.5   |
| VALHALLA (M)     | 11.3 ± 0.2              | 16.6 ± 0.8           | 14.4 ± 1.0            | 14.0 ± 0.4   |

Table 3: **BLEU score on WIT.** Please refer to supplementary material for METEOR score comparisons.

### Results on WMT2014

<table>
<thead>
<tr>
<th>Method</th>
<th>Visual Data</th>
<th>Well-Resourced</th>
<th>Under-Resourced</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>EN $\rightarrow$ DE</td>
<td>EN $\rightarrow$ FR</td>
</tr>
<tr>
<td>Text-Only</td>
<td>–</td>
<td>27.1 ± 0.2</td>
<td>55.0 ± 0.1</td>
</tr>
<tr>
<td>UVR-NMT [74]</td>
<td>Multi30K</td>
<td>27.2 ± 0.2 (28.1)</td>
<td>55.3 ± 0.1</td>
</tr>
<tr>
<td>RMMT [66]</td>
<td>Multi30K</td>
<td>24.5 ± 0.2</td>
<td>52.8 ± 0.1</td>
</tr>
<tr>
<td>VALHALLA</td>
<td>Multi30K</td>
<td>28.0 ± 0.1</td>
<td>56.0 ± 0.1</td>
</tr>
<tr>
<td></td>
<td>WIT</td>
<td>28.0 ± 0.1</td>
<td>56.1 ± 0.1</td>
</tr>
<tr>
<td>VALHALLA (M)</td>
<td>Multi30K</td>
<td>28.0 ± 0.0</td>
<td>56.0 ± 0.1</td>
</tr>
<tr>
<td></td>
<td>WIT</td>
<td>27.9 ± 0.1</td>
<td>56.0 ± 0.2</td>
</tr>
</tbody>
</table>

Table 4: **Results on WMT2014.** UVR-NMT results in brackets are reported by the original paper.
Results

Translation Under Limited Textual Context

**Progressive Masking**: replaces all except the first $K$ words of source sentence

**Visual Entity Masking**: randomly replaces visually grounded phrases

Figure 4: **Evaluation with Progressive Masking**. All results use METEOR scores on Multi30K Test2016 split.

Figure 5: **Evaluation with Entity Masking**. All results use METEOR scores on Multi30K Test2016 split.
Ablation Analysis

- **Discrete vs Continuous Representation**
- **Visual Encoder Parameters**
- **Image Retrieval**

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Discrete Embedding</th>
<th>External Pretraining</th>
<th>Aggregation</th>
<th>EN-DE</th>
<th>EN-FR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP RN-50</td>
<td>×</td>
<td>CLIP</td>
<td>Gating</td>
<td>38.0</td>
<td>58.8</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>×</td>
<td>ImageNet</td>
<td>Gating</td>
<td>38.8</td>
<td>59.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Concatenation</td>
<td>38.3</td>
<td>60.0</td>
</tr>
<tr>
<td>VQGAN VAE</td>
<td>✓</td>
<td>None</td>
<td>Concatenation</td>
<td>39.6</td>
<td>60.5</td>
</tr>
</tbody>
</table>

(a) Discrete and continuous visual encoder backbones, evaluated with Transformer-Small on Multi30K Test2016 split.

<table>
<thead>
<tr>
<th>Encoder Layers</th>
<th>Visual Token Length</th>
<th>EN-DE</th>
<th>EN-FR</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$16^2 = 256$</td>
<td>13.5 ± 7.2</td>
<td>54.3 ± 0.4</td>
</tr>
<tr>
<td>5</td>
<td>$8^2 = 64$</td>
<td>36.3 ± 0.2</td>
<td>60.3 ± 0.2</td>
</tr>
<tr>
<td>6</td>
<td>$4^2 = 16$</td>
<td>39.6 ± 0.3</td>
<td>60.5 ± 0.1</td>
</tr>
</tbody>
</table>

(b) Visual encoder depths, evaluated with Transformer-Small on Multi30K Test2016.

<table>
<thead>
<tr>
<th>Visual Data</th>
<th>Image Retrieval</th>
<th>EN-DE</th>
<th>EN-FR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi30K</td>
<td>×</td>
<td>16.5 ± 0.3</td>
<td>26.2 ± 0.1</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>17.6 ± 0.1</td>
<td>26.9 ± 0.2</td>
</tr>
<tr>
<td>WIT</td>
<td>×</td>
<td>16.6 ± 0.2</td>
<td>26.1 ± 0.3</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>17.7 ± 0.2</td>
<td>26.8 ± 0.0</td>
</tr>
</tbody>
</table>

(c) Training on WMT under-resourced tasks *without* image retrieval.
Ablation Analysis

• Randomized Visual Tokens
  ▸ Decrease performance when used random visual tokens

• Loss Hyperparameters

Figure 7: Influence of loss weights $\gamma_H$ and $\lambda_C$ of (13) on translator performance, measured on Multi30K EN→DE task.
Strengths

• Outperforms other baselines in different settings with extensive experiments
  • In longer sentence the performance gain is much better than short sentence
• No prior work is done on large scale multilingual WIT dataset
  ▸ Proposed a new benchmark in seven language pairs under three settings.
• Generate semantically more informed translation compared to text only models in progressive masking scenario
• A detailed study indicates the visual information improves the performance and translation quality
Weaknesses

• The improvement in non-English language translation is marginal compared to text only baseline in WIT dataset
  ▶ English-centric bias in image text pair
• The concept of hallucination is not novel
• Requires high quality image for the hallucination networks
• Training Complexity:
  ▶ Careful tuning of hyperparameters
  ▶ Sensitive towards consistency loss hyperparameter
Future Works

- Can extend the work with other modalities - video, audio
- More focus on the under represented/ non English language translation tasks
Discussions

• Robustness of the model training since hyperparameter sensitive
• Training complexity
  • Pre-extract VQGAN VAE tokens