Appendix

1 Experimental Details

For both token and character-level translation tasks, we used a maximum area size of 5 for the two layers for Transformer Tiny and Small, and a maximum area size of 4 for the first layer of Transformer Base with Eq.3, and for the first two layers of Transformer Base with Eq.9. For Transformer Big EN-DE, we used a maximum area size of 4 for the first layer with Eq.3, and a maximum area size of 3 for Eq.9. For Transformer Big EN-FR, we used a maximum area size of 3 for the first two layers with Eq.3, and a maximum area size of 4 for the first layer with Eq.9.

For character-level translation tasks, we here used the same dataset, and the experimental strategies as the token-level experiments (see Section 4.1.1) with a few differences to reduce the experiment time because it is significantly slower to train than token-level translation. In particular, we trained all the Transformer Big for 300,000 steps. For Transformer Big EN-DE, we could use a larger batch size that amounts to approximately 32,000 characters. All the LSTM models use the same batch size that amounts to 164,000 characters for 50,000 steps.

2 Additional Experimental Results

We evaluated the approach of area feature combination (Eq.9) on character-level translation tasks as well. It performed on par with the basic form of area attention (Eq.3). In particular, it outperformed the basic form (see details in Table 3), with statistical signficance, on Transformer Tiny EN-FR ($BLEU = 12.91$) and Transformer Small EN-FR ($BLEU = 21.93$) and EN-DE ($BLEU = 14.5$), which seem to imply that when the basic area attention helps, the method of area feature combination could bring further improvements.

We also explored the approach of using normalized sigmoid (Shen Lee, 2016; Rei Søgaard, 2018) as the activation function of multi-head attention in Transformer. The quick experiments with Transformer Tiny and Small by replacing softmax with normalized sigmoid led to poor results, which deserves further investigation.

3 Related Algorithmic Details for Integral Images

Summed area table is based on an integral image, I, which can be efficiently computed in a single pass of the memory (see Equation 1). Here let us focus on the area value calculation for a 2-dimensional memory because a 1-dimensional memory is just a special case with the height of the memory grid as 1.

$$I_{x,y} = v_{x,y} + I_{x,y-1} + I_{x-1,y} - I_{x-1,y-1} \tag{1}$$

where x and y are the coordinates of the item in the memory. With the integral image, we can calculate the key and value of each area in constant time. The sum of all the vectors in a rectangular area can be easily computed as the following (Equation 2).

$$v_{x1,y1,x2,y2} = I_{x2,y2} + I_{x1,y1} - I_{x2,y1} - I_{x1,y2} \tag{2}$$

where $v_{x1,y1,x2,y2}$ is the value for the area located with the top-left corner at (x_1, y_1) and the bottom-right corner at (x_2, y_2). By dividing $v_{x1,y1,x2,y2}$ with the size of the area, we can easily compute $\mu_{x1,y1,x2,y2}$. Based on the summed area table, $\sigma^2_{x1,y1,x2,y2}$ (thus $\sigma_{x1,y1,x2,y2}$) can also be computed at constant time for each

area (see Equation 3), where $I^2_{x,y} = v^2_{x,y} + I^2_{x,y-1} + I^2_{x-1,y} - I^2_{x-1,y-1}$, which is the integral image of the element-wise squared memory.

$$\sigma^2_{x1,y1,x2,y2} = \frac{I^2_{x2,y2} + I^2_{x1,y1} - I^2_{x2,y1} - I^2_{x1,y2}}{(x2 - x1) \times (y2 - y1)} - \mu^2_{x1,y1,x2,y2} \tag{3}$$

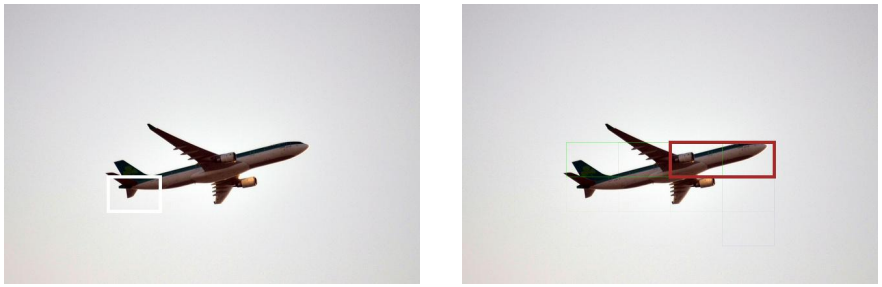
The core component for computing these quantities is to be able to quickly compute the sum of vectors in each area after we obtain the integral image table I for each coordinate $[x, y]$, as shown in Equation 1 and 2. The Pseudo code for performing these are presented in the paper, which is based on efficient Tensor operations (see Algorithm 1 and 2).

4 Attention Visualization

To understand the behavior of area attention, we analyzed the attention distribution of the trained Transformer models for both image captioning (see Figure 1) and character-level machine translation tasks (see Figure 2). For both visualizations, we analyzed the encoder self attention. For image captioning, it is the relationship between a fixed-sized query cell from a 8x8 grid to a varying-sized area that can involve multiple adjacent grids on an image. For character-level machine translation, it shows how a query character attends to a group of adjacent characters in the same sentence.

(a) a bird sitting on top of a tree branch

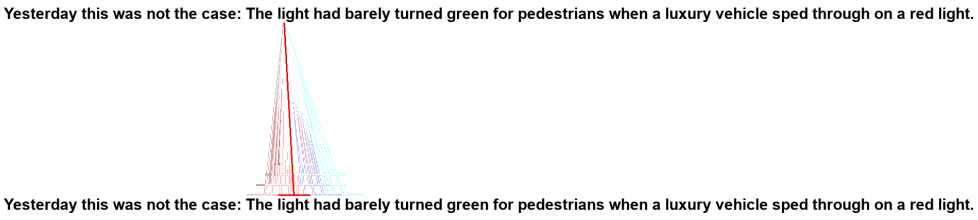
(b) a small yellow bird perched on a branch



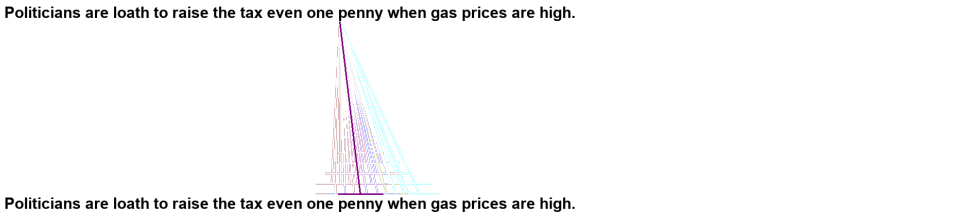
(c) an airplane is flying in the sky

(d) a man flying through the air while riding a skateboard

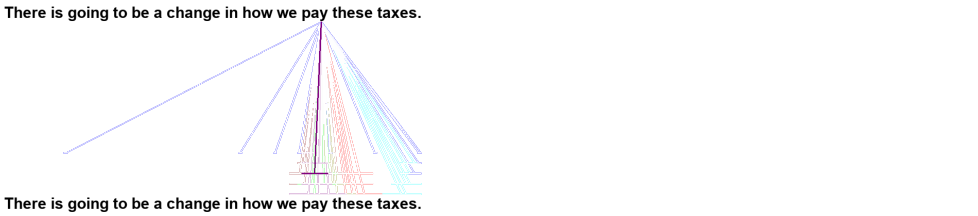
Figure 1: The images on the left show the query grid and the images on the right are attended areas. The top attended area is highlighted with the color of the corresponding attention head.



(a)

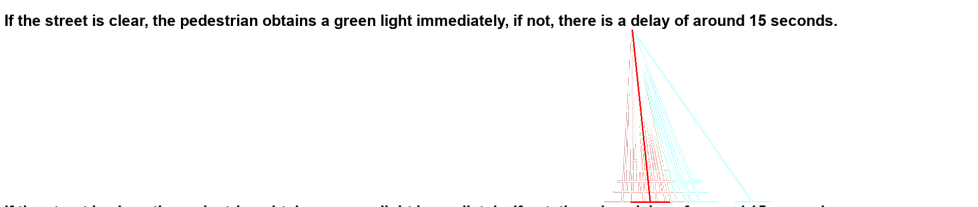


(b)



(c)

(d)



(e)

Figure 2: Examples of self-attention (using area attention) for the first layer of a 6-layer Transformer encoder, during character-level EN-DE translation tasks. For each example in Figure 2, the top row shows the query character and the bottom row shows the attended range of characters. The color indicates one of the 8 heads for the multi-head attention while the intensity shows the probability of the attention.