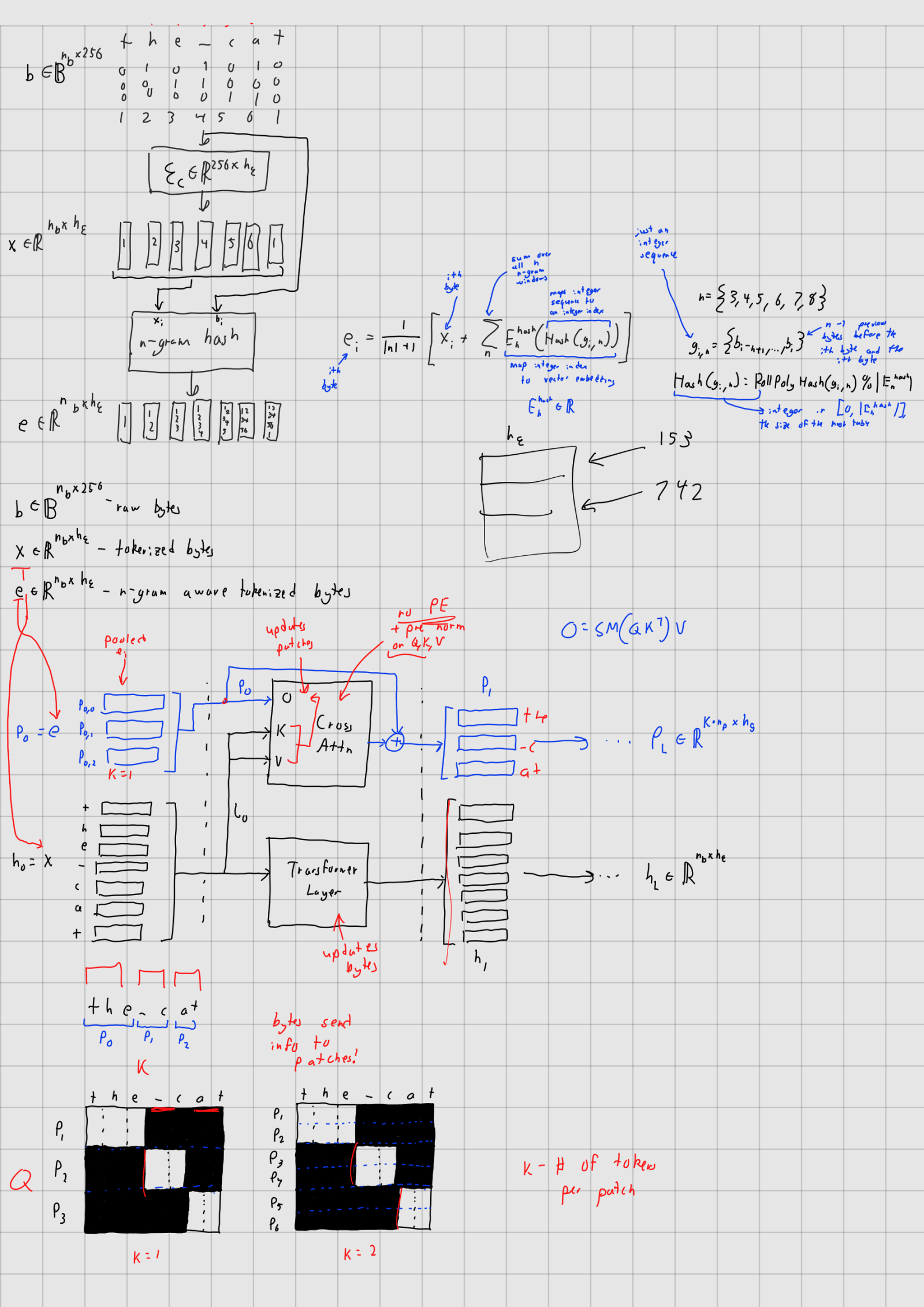
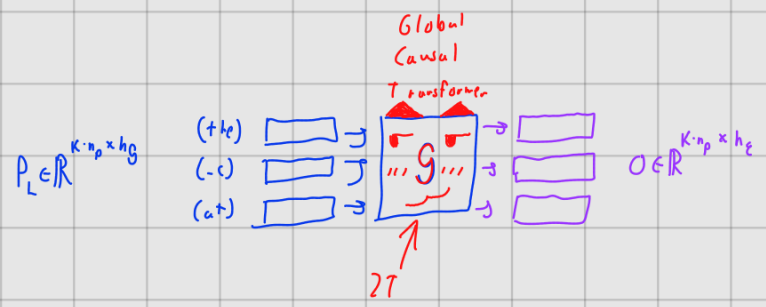
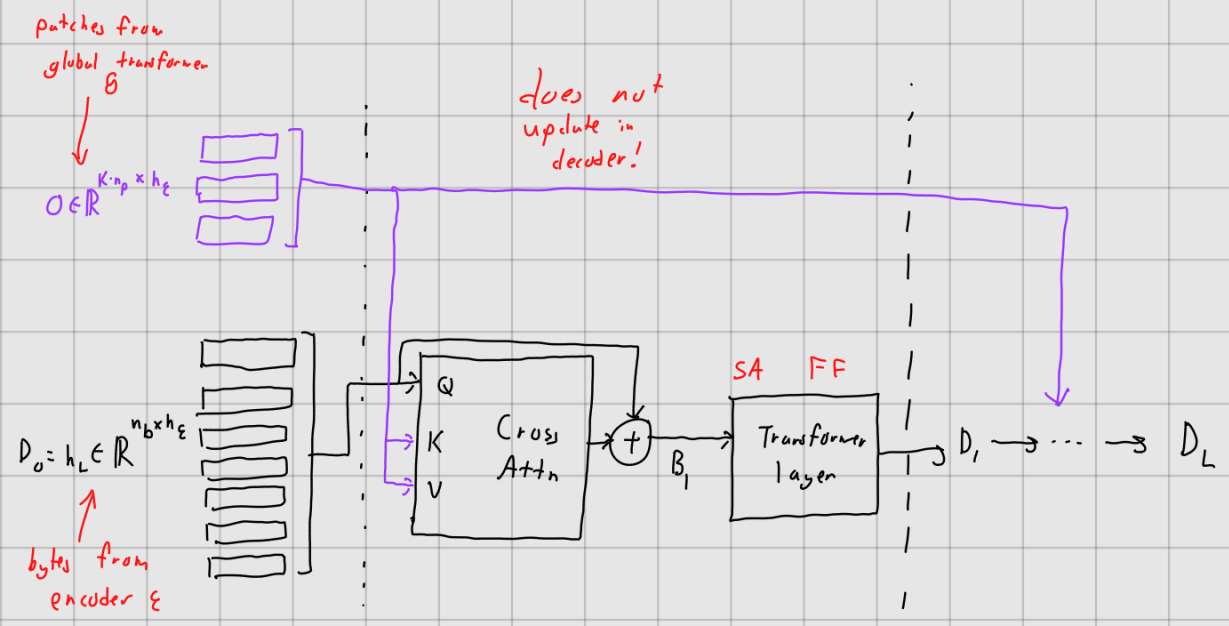

$b \in \mathbb{B}^{n_b \times 256}$
t h e _ c a t
$\mathcal{E}_c \in \mathbb{R}^{256 \times h_\mathcal{E}}$
$x \in \mathbb{R}^{n_b \times h_\mathcal{E}}$
x_i
b_i
n-gram hash
$e \in \mathbb{R}^{n_b \times h_\mathcal{E}}$
$e_i = \frac{1}{|n|+1}\left[x_i + \sum_n E_n^{hash}(Hash(g_{i,n}))\right]$
i-th byte
sum over all n n-gram windows
map integer sequence to an integer index
map integer index to vector embeddings
just an integer sequence
$n = \{3, 4, 5, 6, 7, 8\}$
$g_{i,n} = \{b_{i-n+1}, \ldots, b_i\}$
n-1 previous bytes before the i-th byte and the i-th byte
$Hash(g_{i,n}) : RollPolyHash(g_{i,n}) \% |E_n^{hash}|$
integer in $[0, |E_n^{hash}|]$ the size of the hash table
$E_n^{hash} \in \mathbb{R}$
$h_\mathcal{E}$
153
742
$b \in \mathbb{B}^{n_b \times 256}$ - raw bytes
$x \in \mathbb{R}^{n_b \times h_\mathcal{E}}$ - tokenized bytes
$e \in \mathbb{R}^{n_b \times h_\mathcal{E}}$ - n-gram aware tokenized bytes
pooled e_i
$p_0 = e$
$p_{0,0}$
$p_{0,1}$
$p_{0,2}$
K=1
updates patches
p_0
Q
K
V
Cross Attn
RoPE + pre norm on Q, K, V
$O = SM(QK^T)V$
p_1
+L
-c
at
$p_L \in \mathbb{R}^{K \cdot n_p \times h_g}$
$h_0 = x$
+ h e _ c a t +
l_0
Transformer Layer
updates bytes
h_1
$h_L \in \mathbb{R}^{n_b \times h_\mathcal{E}}$
t h e _ c a t
p_0 p_1 p_2
K
bytes send info to patches!
t h e _ c a t
p_1
p_2
p_3
Q
K=1
t h e _ c a t
p_1
p_2
p_3
p_4
p_5
p_6
K=2
K - # of tokens per patch





Sampling:

```
cur_seq = "the_cat"
for byte in num_bytes_to_pred:                    # can probably speed up with KV cache for each byte
    patch_mask = get_patches(cur_seq)
    # , e = encode_and_hash(cur_seq)              # only needs to be redone for newest byte
    p = pool_patches(e_i)                         # rerun for last patch
    h = X
    for enc_blk in encoder:                       # update patches and bytes
        p, h = enc_blk(p, h)
    for global_blk in Global:                     # update patches
        p = global_blk(p)
    O = p
    for dec_blk in decoder:                       # update bytes
        h = dec_blk(h, O)
    new_byte = sample(SM(h[-1]))                  # add new byte to seq
    cur_seq = cur_seq + new_byte
```

can probably improve via KV cache. Note the last token cannot be cached until the patch is complete.