**BBM480   
PROGRESS REPORT**

**2**

(27.04.2025)

Tamer Yeşildağ

Eray Taymaz

**Model Architecture**

Our current model architecture is aligned with Sign2GPT’s own model, down to every hyper parameter and module.

| Component | Sign2GPT Paper | Code |
| --- | --- | --- |
| **Backbone** | DINO-V2 ViT-S/14, frozen except for LoRA adapters on the top 3 blocks (qkv, proj, MLP). LoRA rank 4, α=4. | Loads DINOv2 ViT-S/14 via torch.hub, freezes all weights, injects LoRALinear into the last 3 blocks with r=4r=4r=4, α=16 (scale =4). |
| **Sign Encoder** | 4-layer spatio-temporal Transformer: local attention (window=7), temporal downsampling after layer 2, hidden=512, heads=8, FF=2048. | SignTransformerEncoder uses 4 layers, window\_size=7, downsampling after layer 1 index (i.e. 2nd), and the same dims & heads. |
| **Pretraining (Pseudo-Gloss)** | Prototype-driven pretraining: generate word‐level “pseudo‐glosses” from text, initialize UUU prototypes via fastText, train the sign encoder to align features to prototypes using a joint temporal/class softmax and BCE loss. | PretrainingModule encapsulates this: loads prototypes.pkl, defines learnable τ’s, computes the same temporal & prototype softmax, and BCE loss. |
| **Positional Encoding** | Sinusoidal positions are added before mapping sign features into the decoder to reintroduce temporal order. | The code inserts positional encodings inside the sign encoder itself. Slight placement difference, but both use classic sin/cos embeddings. |
| **Training Hyperparams** | Pretraining and translation both use AdamW (lr 3e-4, wd 1e-3), 100 epochs, gradient clipping 1.0, one-cycle cosine scheduler with 5% warmup, bfloat16 & FlashAttention-v2. | Matches exactly for pretraining: AdamW lr 3e-4, weight\_decay 1e-4, 100 epochs, clip\_norm 1.0, OneCycleLR with pct\_start=0.05, cosine annealing. |

**Problems**

1-**Google Colab GPU Memory Constraints:** Due to GPU memory constraints on Google Colab, we had to reduce the batch size from 8 to 2.

| OutOfMemoryError: CUDA out of memory. Tried to allocate 678.00 MiB. GPU 0 has a total capacity of 39.56 GiB of which 450.88 MiB is free. Process 23079 has 39.11 GiB memory in use. Of the allocated memory 37.90 GiB is allocated by PyTorch, and 734.65 MiB is reserved by PyTorch but unallocated. If reserved but unallocated memory is large try setting PYTORCH\_CUDA\_ALLOC\_CONF=expandable\_segments:True to avoid fragmentation. See documentation for Memory Management (https://pytorch.org/docs/stable/notes/cuda.html#environment-variables) |
| --- |

2-**Long Train Time**: Even though we use A100 which is the most powerful gpu available in google colab, training takes almost 3 hours per epoch. Training the model for 100 epochs like in the paper is not possible because google colab terminates sessions longer than 24 hours.  
  
 Figure: tqdm for 1 epoch  
  
3-**“Device Side Assert” Error**: We are encountering a CUDA error during the validation step. It is challenging to trace back this error. We are still trying to find the root cause of this error.  
  
