A Statistics of pseudo data

Table 4 shows the statistics of pseudo ST data in terms of the number of sentences and the corresponding hours of speech recordings.

	ru	zh	pt	fa	et	mn	nl	tr	ar
Num of sentences	14.9 K	17.5 K	16.0 K	26.0 K	2.8 K	0.3 K	30.3 K	26.0 K	28.0 K
Duration (hours)	20.5	24.0	17.6	27.8	5.2	0.5	37.3	25.3	31.8
	wv	lv	ta	ja	id	sl	cy	Total	
Num of sentences	6.3 K	1.1 K	41.7 K	6.5 K	5.0 K	1.0 K	7.6 K	231.1 K	
Duration (hours)	7.3	0.9	75.4	8.8	7.8	1.0	10.9	302.2	

Table 4: Statistics of pseudo ST data

B Breakdown of results on CoVoST 2 evaluation set

The ASR and MT tasks are used as auxiliary tasks in the training of our ComSL models for the ST task. In the inference stage, we can also conduct these two tasks to assess the effectiveness of the unified speech-text representations learned by our approaches, in addition to ST task.

B.1 ST task

The breakdown of BLEU scores in the different configurations on each of the 21 language pairs is listed in Tabel 5. The corresponding average BLEU scores are shown in Tabel 1.

xx-en	High-resource				Mid-resource					Low-resource	
	fr	de	es	ca	fa	it	ru	pt	zh	tr	ar
Whisper Large (1.6B)	38.3	35.8	40.7	33.3	20.2	37.2	42.0	51.7	18.0	30.9	38.2
Whisper Large + mBART-50 (2.2B)	38.8	37.0	40.7	33.0	16.8	36.5	49.0	49.1	21.5	32.7	37.0
ComSL Medium (0.9B)	38.6	35.6	40.2	35.0	22.4	36.2	49.2	49.7	20.5	32.5	40.9
ComSL Large (1.3B)	38.8	36.0	40.4	35.3	22.4	36.6	49.2	49.9	21.4	33.6	41.4
xx-en	Low-resource										
	et	mn	nl	sv	lv	sl	ta	ja	id	cy	
Whisper Large (1.6B)	12.8	0.7	41.5	45.6	15.5	24.6	4.0	26.1	49.4	17.8	
Whisper Large + mBART-50 (2.2B)	16.3	0.4	39.9	44.7	21.4	25.0	4.1	23.0	45.5	27.0	
ComSL Medium (0.9B)	18.4	2.4	38.7	42.8	18.8	28.8	5.0	20.3	46.3	24.7	
ComSL Large (1.3B)	19.2	2.9	39.7	43.4	21.3	31.6	5.0	21.3	46.6	24.5	

Table 5: The BLEU scores of ST on CoVoST 2 test set.

B.2 ASR task

The performance of multi-lingual ASR in terms of WER is shown in Tabel 6. There is no available standard for multi-lingual text normalization or word segmentation for some languages, such as Arabic. The use of different tokenizers by Whisper and mBART also affects the WER results. Our WER measurement procedure involves: detokenize → remove punctuation → split word (optionally) → calculate WER. As a result, these WER numbers cannot be directly referred to compare with the numbers in other publications if they exist.

	High-resource				Mid-resource					Low-resource	
	fr	de	es	ca	fa	it	ru	pt	zh	tr	ar
Whisper Large (1.6B)	0.10	0.08	0.06	0.09	0.40	0.09	0.06	0.06	0.12	0.12	0.33
ComSL Medium (0.9B)	0.10	0.10	0.08	0.07	0.33	0.12	0.13	0.08	0.22	0.24	0.82
ComSL Large (1.3B)	0.10	0.10	0.08	0.07	0.33	0.12	0.12	0.08	0.22	0.22	0.74
	Low-resource										
	et	mn	nl	sv	lv	sl	ta	ja	id	cy	
Whisper Large (1.3B)	0.42	0.97	0.09	0.12	0.34	0.28	0.25	0.12	0.11	0.40	
ComSL Medium (0.9B)	0.35	0.76	0.15	0.20	0.46	0.37	0.16	0.45	0.25	0.29	
ComSL Large (1.3B)	0.33	0.74	0.15	0.19	0.43	0.36	0.16	0.43	0.24	0.29	

Table 6: The WERs of ASR on CoVoST 2 test set.

B.3 MT task

Tabel 7 lists the BLEU scores for machine translation using ground-truth transcription as input on each of the 21 language pairs.

	High-resource				Mid-resource					Low-resource	
	fr	de	es	ca	fa	it	ru	pt	zh	tr	ar
mBART-50 (0.6B)	46.1	40.7	45.3	36.4	27.6	41.8	52.1	52.4	25.8	36.9	48.3
ComSL Medium (0.9B)	46.4	41.2	45.8	37.5	28.2	42.1	51.7	52.3	24.9	37.2	47.2
ComSL Large (1.3B)	46.4	41.1	45.7	37.4	28.2	42.2	51.7	52.5	24.8	37.4	47.6

	Low-resource									
	et	mn	nl	sv	lv	sl	ta	ja	id	cy
mBART-50 (0.6B)	27.5	9.1	43.2	52.9	33.5	38.9	6.4	25.1	53.4	54.0
ComSL Medium (0.9B)	28.3	9.8	43.4	52.0	34.0	39.5	6.1	24.8	54.6	39.5
ComSL Large (1.3B)	28.2	9.8	43.7	52.6	33.9	40.5	6.2	24.8	53.1	40.0

Table 7: The BLEU scores of MT on CoVoST 2 test set.

C Experimental Details

We use an Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $weight\_decay = 0.1$ and a polynomial decay scheduler for all our experiments. For finetuning mBART-50 on CoVoST 2, we use the scheduler with a learning rate of 2e-5 and a warmup of 2.5k steps. The mBART-50 model is finetuned for 5 epochs on the training set and then functions as the regularization model during ComSL training. We set the attention dropout to 0.1 and the other dropout to 0.3. The checkpoint trained by 2 epochs is used for initializing ComSL language blocks. For ComSL training, we use a scheduler with a learning rate of 2e-5 and a warmup of 5k steps. The weights we use for different losses are $w_{asr} = 0.35$, $w_{st} = 0.35$, $w_{mt} = 0.2$, $w_{CML} = 0.1$, $w_{ERM} = 0.1$, $\lambda_s = 0.8$ for DDM, $\lambda_t = 0.2$ for MT regularization. We set the attention dropout to 0 and the other dropout to 0.1 for language blocks, and no dropout for speech blocks (following Whisper). During the first third of the training procedure, we fix the parameters of the speech blocks. We use PyTorch Lightning as our code framework.

The comparisons with previous works We compare our CML method with other methods and show the results in Table 3. The baseline models, where these methods were implemented, vary from ours in terms of experimental configuration, model size, and training data. It makes direct comparison difficult. Instead, we just implemented them based on our model and kept the same architectures and hyperparameters as theirs. So it may not be optimal for these methods.

- **MML** Modality Matching Loss (MML) was employed in Maestro and USM, i.e., an L2 loss between the speech and upsampled text embeddings. Since we do not have an RNNT model, we train an external CTC model for forced alignment at the sub-word level. We replicate the initially learned text embeddings to match the duration of the speech embedding using this alignment information. We skip the refiner and directly calculate the L2 loss on these two aligned embeddings.
- **ConST** ConST calculates the mean pooling on the speech or text embedding sequence as the representation of the sentence and adds a contrastive loss on paired speech and text representation.
- **WACO** This method is an improvement over CONST in that it performs mean-pooling on the word level and adds contrastive loss on paired representations of speech and text. We leverage a force-aligner that is modified from whisper-timestamped[6] to conduct forced alignment on the training set.

[6]https://github.com/linto-ai/whisper-timestamped