Interspeech 2024
1-5 September 2024, Kos, Greece

MSRS: Training Multimodal Speech Recognition Models from Scratch with Sparse Mask Optimization

Adriana Fernandez-Lopez[1], *Honglie Chen*[1], *Pingchuan Ma*[1,2], *Lu Yin*[3], *Qiao Xiao*[4], *Stavros Petridis*[1,2], *Shiwei Liu*[5], *Maja Pantic*[1,2]

[1]Meta AI, UK [2]Imperial College London, UK [3]University of Surrey
[4]Eindhoven University of Technology [5]University of Oxford

{afernandezlopez,honglichen,pingchuanma}@meta.com

Abstract

Pre-trained models have been a foundational approach in speech recognition, albeit with associated additional costs. In this study, we propose a regularization technique that facilitates the training of visual and audio-visual speech recognition models (VSR and AVSR) from scratch. This approach, abbreviated as **MSRS** (Multimodal Speech Recognition from Scratch), introduces a sparse regularization that rapidly learns sparse structures within the dense model at the very beginning of training, which receives healthier gradient flow than the dense equivalent. Once the sparse mask stabilizes, our method allows transitioning to a dense model or keeping a sparse model by updating non-zero values. MSRS achieves competitive results in VSR and AVSR with 21.1% and 0.9% WER on the LRS3 benchmark, while reducing training time by at least 2×. We explore other sparse approaches and show that only MSRS enables training from scratch by implicitly masking the weights affected by vanishing gradients.

Index Terms: speech recognition, sparse mask optimization, sparse regularization, sparse networks.

1. Introduction

Automatic Speech Recognition (ASR), Visual Speech Recognition (VSR) and Audio-Visual Speech Recognition (AVSR) systems use various input sources (audio, video, and audio-visual) to map spoken language into written text. The success of these systems is due to advanced deep neural architectures [1–4] and large-scale audiovisual datasets [4, 5], where performance usually scales with both model size and training data. However, training massive deep networks is challenging and comes with well-known hurdles such as overfitting or vanishing/exploding gradients. Regularization strategies such as dropout, batch normalization, warm-up, gradient clipping, and data augmentation have been adopted but are not always sufficient for training deep VSR/AVSR models from scratch [2, 3]. Current VSR/AVSR methods typically follow a two-stage curriculum learning approach [6], involving initial pretraining on a small subset of data ($1-5$ cycles [7, 8]), followed by continued training on a large-scale dataset [7, 9]. Another line of work achieves a better convergence based on sub-words learning schemes, where training samples are cropped out with frame-word boundaries [10]. In some cases, auxiliary tasks are also incorporated [2, 8]. Despite their success in training VSR/AVSR models, pretraining can be resource-intensive as the model and dataset size grows [3, 11].

To facilitate training VSR/AVSR models from scratch, we investigate their gradient norm when trained from scratch and observe a significant phenomenon of gradient explosion. This phenomenon can be addressed by clipping the gradient to an appropriate value. However, as the network becomes deeper and more complex, the gradient also vanishes. Figure 1 illustrates this problem by showing the gradient norm of the first feed-forward layer in the VSR encoder. When no clipping is applied (grey line), the gradients explode, while only clipping (blue line) fails to solve vanishing problems. To address this issue, we investigate optimization strategies that enable the training of deeper high-capacity models. One approach is to optimize the Transformer/Conformer blocks, as suggested by various studies [12–15]. Among these methods, LayerScale [15] stands out for regularizing deeper vision transformers. LayerScale incorporates a learnable diagonal matrix, initialized close to 0, to the output of each residual block to enhance the training dynamics. Another possible approach involves pruning weights to reduce the number of hidden units while maintaining the dense connectivity of the model [16–19]. For instance, the Gradual Magnitude Pruning (GMP) algorithm [16] starts from a randomly initialized dense model and gradually prunes the smallest magnitude weights until it achieves the target sparsity. Encouragingly, their experiments demonstrate that large-sparse models consistently surpass small-dense models in various tasks with the same memory footprint and minimal performance loss.

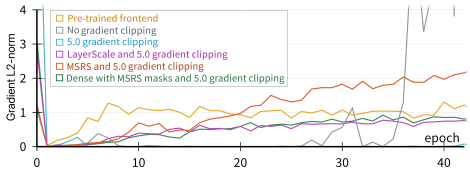

Figure 1: *A close-up view of half of VSR training process. L2 gradient norm of the first feed-forward layer in the encoder of multiple models trained with various regularization strategies. Assume uniform initializations, if no pre-training is specified.*

In this work, we introduce a mask optimization technique coined **MSRS** (Multimodal Speech Recognition from Scratch), which rapidly identifies a sparse topology within the dense model. This sparse topology receives a more stable gradient flow, and hence, stabilizes the training process, enabling training very large speech models without the need for any pretraining steps. Once the sparse mask becomes stable (within a few epochs), our approach allows for condensing to a dense model or maintaining a sparse model by only updating the non-zero values of the mask. Following, we outline our contributions as: (i) we present MSRS for efficient training of VSR/AVSR models from scratch; (ii) we investigate our approach in the limited-data regime and find that MSRS allows for training VSR models in situations where data is limited (around 90 hours). This makes it possible to extend to languages with scarce data available; (iii) we compare our method with numerous sparse training approaches, none of which can train large-scale VSR models from scratch without a pretrained frontend. We study the topological differences of their subnetworks and provide in-

10.21437/Interspeech.2024-1153

sight into why MSRS converges. We find that MSRS explicitly masks the weights that experience vanishing gradients to enhance training, whereas other methods that decouple initialization from the training process do not; (iv) additionally, we delve into an investigation that draws connections to LayerScale. This analysis undercores the complementary nature of MSRS and LayerScale in improving the gradient flow of deep speech models by regulating the width and depth of the model, respectively; (v) finally, the effectiveness of MSRS is backed up with strong empirical results on VSR and AVSR tasks. For instance, despite being trained from scratch, our condensed VSR model reports competitive results on the LRS3 benchmark, while reducing training time more than **2**×. Similarly, our condensed AVSR model outperforms the current state-of-the-art (SoTA) [1] by more than 3% WER on highly corrupted speech, without using pretraining stages and under lower 16-float precision.

Please note that the primary goal of this work is not to achieve efficiency through sparsity, but rather to explore the under-explored benefits of irregular sparse patterns in enabling the training of VSR/AVSR models from scratch.

2. Approach

For a speech recognition task, such as VSR, ASR, or AVSR, we have a model $f(\cdot;\theta)$ with model parameters θ that inputs a video, audio, or audio-visual sequence x and outputs the corresponding transcription y from a dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ with N sequences. This model is composed of a frontend for each modality, a multi-layer perceptron to combine the features from multiple domains (for AVSR), a Conformer encoder, and a Transformer decoder for joint training using connectionist temporal classification ($\mathcal{L}_{ctc}$) and attention ($\mathcal{L}_{att}$):

$$\mathcal{L}(\theta) = \mathcal{L}_{att} + \gamma\mathcal{L}_{ctc} \tag{1}$$

Consider a scenario where the model $f(\cdot;\theta)$ has millions parameters, making training complex and difficult to train from scratch [2, 3]. To address this issue, we seek a mask $m \in \{0,1\}^p$ that can progressively shrink the width of the model $f(\cdot;\theta \odot m)$, resulting in a smaller and easily trainable model starting from the initial weights θ_0, where p represents the number of model parameters and $\odot$ is the element-wise product.

2.1. Mask optimization

Inspired by [20, 21], we simultaneously search for an optimal model structure $m \in \{0,1\}^p$ and its corresponding weights $\theta \in \mathbb{R}^p$, where θ is continuous in $\mathbb{R}$ and m is discrete in $\{0,1\}$:

$$m_* = \underset{m}{\operatorname{argmin}}(\min_{\theta} \mathcal{L}(m,\theta), \mathcal{D}) \tag{2}$$

$$\phi_0 = (\ln(|\theta_0| + \varsigma)/2 + 1) \cdot \rho + \mu \tag{3}$$

This fact complicates optimization since $\mathcal{L}$ is not differentiable with respect to m, and standard gradient descent techniques cannot be directly applied. Therefore, we use a fully continuous approximation of that problem. Concretely, the sigmoid function is used to approximate the mask in continuous space: $m \to m_l = \sigma(l\phi)$, where ϕ_0 is initialised near zero according to eq. (3) *i.e.,* small absolute values are mapped to negative and larger ones to positive. l controls the *sharpness* of the approximation, *i.e.,* the larger, the closer to the binary function. The discrete optimization in eq. (2) becomes continuous in eq. (4):

$$\phi_* = \underset{\phi}{\operatorname{argmin}}(\min_{\theta} \mathcal{L}(m_l,\theta), \mathcal{D})$$
$$\text{subject to} \quad m_l = \sigma(l\phi) = \frac{1}{1+e^{-l\phi}}, \quad l \gg 1 \tag{4}$$

In this scenario, m_l is differentiable for all $l < \infty$, but when $\phi \neq 0$ and $l \to \infty$, $\nabla_\phi m_l = \sigma(l\phi)\cdot(1-\sigma(l\phi)) \to 0$, which leads to vanishing gradients. To deal with that we consider using two-temperature minimization [22], *i.e.,* use a larger value in the sigmoid during forward propagation $\overrightarrow{l}$ and a smaller during backward propagation $\overleftarrow{l}$. This results in gradient updates:

$$\phi_{i+1} \leftarrow \phi_i - \eta_i \nabla_{m_{\overrightarrow{l}}} \mathcal{L}(z; m_{\overrightarrow{l}}, \theta) \nabla_\phi m_{\overleftarrow{l}} - \lambda \mathbf{1}_p \tag{5}$$

$$\theta_{i+1} \leftarrow \theta_i - \alpha_i \nabla_\theta \mathcal{L}(z; m_{\overrightarrow{l}}, \theta) \tag{6}$$

where η_i and α_i are learning rates, $z = (x, y)$ refers to input-output sequences, $\mathbf{1}_p = (1, 1, ..., 1) \in \mathbb{R}^p$, and λ is a hyperparameter that controls the pruning speed. The penalization term in eq. (7) encourages the entries of ϕ to be negative, leading to sparser masks.

$$\mathcal{L}(m_l,\theta) = \mathcal{L}_{att} + \gamma\mathcal{L}_{ctc} + \lambda \mathbf{1}_p^\top \phi \tag{7}$$

After T epochs, a near-optimal binary mask can be obtained from ϕ_T by setting all positive values to 1 and all negative values to 0, where $\mathbb{1}$ is the indicator function:

$$m_* = \mathbb{1}[\,\phi_T > 0] = \begin{cases} 1 & \phi_T \geq 0 \\ 0 & \phi_T < 0 \end{cases} \tag{8}$$

2.2. Regularization early in training

Algorithm 1 Multimodal Speech Recognition from Scratch

Inputs: model $f(\cdot, \theta_0, \phi_0)$; initial model θ_0 and mask ϕ_0 parameters; training data $\mathcal{D}$; hyperparameters $\overrightarrow{l}$, $\overleftarrow{l}$, ϵ and λ; learning rates η and α; number of epochs T; $dense$ flag.
Set step j=1
while *stopping criteria* $< \epsilon$ or $j \leq T$ **do** ▷ *stopping criteria* compares the sparsity between consecutive binary masks
  Update ϕ_j following eq. (5)
  Update θ_j following eq. (6)
  Increase step j
Get binary mask $m_* = \mathbb{1}[\,\phi_j > 0]$ and model weights θ_j
Re-start η and α and train from model weights $\theta_0 = \theta_j \odot m_*$
for i=1,...,T **do**
  if $dense$ **then** ▷ *All weights are adjustable*
    Update $\theta_{i+1} \leftarrow \theta_i - \alpha_i \nabla_\theta \mathcal{L}(z; \mathbf{1}_p, \theta)$
  else ▷ *Only non-zero weights are adjustable*
    Update $\theta_{i+1} \leftarrow \theta_i - \alpha_i \nabla_\theta \mathcal{L}(z; m_*, \theta)$
return θ_T, m_*

Deep networks can have issues with vanishing or exploding gradients, which can make it difficult for them to converge effectively. This problem is particularly pronounced at the start of training, where regularization is essential for success. However, we have discovered that by using the appropriate λ parameter, the mask gradually converges to a near-optimal sparsity early in the training process ($\sim$ 3 epochs) and remains stable for the remainder of the training. In fact, the mask learns to zero-out those weights that cause convergence issues, resulting in a narrower model that can benefit from depth. This is helpful because optimizing deep models and their masks at the same time throughout the entire training process can be computationally expensive. The algorithm for jointly optimizing the model and mask early in training is shown in Algorithm 1. After that, we re-start the training on the regularized model using the generated mask and the model weights as initialization. At this step, we can choose whether to go back to a dense model or stay sparse by only updating the weights that correspond to the non-zero values of the mask. We reset the learning rate scheduler because models recover from pruning more easily with larger learning rates and struggle to recover with smaller ones [23].

3. Experimental setup

Dataset and evaluation. We use the LRS3 dataset [24], which is the largest publicly available audio-visual dataset in English and a standard benchmark for VSR/AVSR research. The dataset contains 150,498 utterances for training (438 hours) and 1,321 utterances for testing (0.9 hours). Additionally, following previous works [1, 25], we use VoxCeleb2 [26] and AVSpeech [27] audio-visual datasets for training, resulting in a total of 1,307 and 1,323 hours. We follow previous works [28] and report results using Word Error Rate (WER).

Pre-processing. We follow the pre-processing steps used in prior works [2, 8] to crop a 96×96 region centred around the mouth and then transform each frame into a greyscale image.

Data augmentation. We apply horizontal flipping, random cropping, and adaptive time masking [8] to all our models. The masks we use are proportional to the length of the utterance and have a maximum masking length of up to 0.4 seconds.

VSR architecture details. VSR baselines follow the architecture in [25], which includes a small and a large model with 64 and 250 million parameters. Models consist of a 3D convolutional layer, a 2D ResNet-18 [29], a Conformer encoder, a projection CTC layer, and a Transformer decoder. The small/large model has a 12-layer Conformer encoder with 256/768 inputs, 2,048/3,072 feed-forward dimensions, 4/12 attention heads, and a decoder with 6-layer Transformer with same dimensions and number of heads as the corresponding encoder.

AVSR architecture details. AVSR models are composed of a ResNet frontend for each modality [7], a multi-layer perceptron for early fusion of multi-domain features, a single Conformer encoder, and a Transformer decoder. We use 2 models with 268/703 million parameters. The encoder consists of 12/20 layers with 768/1,024 inputs, 3,072/4,096 feed-forward dimensions, and 12/16 attention heads. The decoder is a 6/9-layer Transformer with the same dimensions and number of heads as the encoder. Audiovisual frontend outputs are concatenated and fed into a 2-layer MLP with 8,192 units and 768/1,024 outputs.

Training strategies. The output is decoded using 5,000 unigram subwords. Two AdamW optimizers ($\beta_1 = 0.9$, $\beta_2 = 0.98$) are used with a cosine learning rate scheduler and gradient clipping at 5.0. The peak learning rates are set to 6e-4 for single dataset training and 1e-3 for multiple sets. The model is initially warmed up for up to 5 epochs of the j epochs of joint training, and then warmed up again for 5 epochs during the subsequent 75 epochs of single training. Here, j is the number of joint training epochs that are required for mask stabilization (j=3 if not otherwise specified). No language model is included.

Mask hyper-parameters. We set the initial mask weights ϕ_0 to be very close to 0 with μ =1e-3, ρ =5e-4 and ς =1e-8. This allows both positive and negative weights to easily fluctuate at the beginning of training. We experimentally set up λ = 2e-10, ϵ = 0.01, and hyperparameters $\overrightarrow{t}$=1e5 to approximate the binary function and $\overleftarrow{t}$=1 to avoid vanishing gradients.

4. Results

4.1. End-to-end training of VSR models from scratch

To compare the capability of VSR models trained from scratch, we conduct experiments on the LRS3 dataset and report the results in Table 1. We observe that overall, both small and large dense models face convergence issues when the weights are randomly initialized. Comparing the small models trained from scratch, we show that using LayerScale [15] or MSRS results in a WER from 100 % to 42.8 % and 48.5 %, respectively. The gain is likely due to a healthier gradient flow compared to the dense equivalent. Unlike LayerScale which adds extra parameters to the model, we further show that MSRS can benefit from sparse computation. For instance, when almost 40% of the parameters are zeroed out, a gain of 1.2 % is observed. For large models, in contrast to previous works that solve the optimization issues by using pre-trained weights [1–3, 7, 25], MRSR results in a comparable WER of 21.1 %, while speeding up training by at least 2×, depending on the pretraining strategy [8]. Similarly, when 40% of the parameters are zeroed out, there is only an increase of 2.8 % in WER. This result is in line with our hypothesis that while resource-constrained settings may have limited capacity, large models, which are often challenging to train, can especially benefit from these regularization methods. Furthermore, a reduction of 0.7 % in WER is observed when combining both LayerScale and MSRS. This indicates that these two regularization strategies complement each other, where LayerScale almost eliminates the residual connections of the encoder at the start of training, limiting its depth, while MSRS quickly in training discovers sparse structures within the dense model, resulting in relatively smoother gradient flow compared with the dense counterpart.

Table 1: *VSR models under different initializations (θ_0) and regularizations on the LRS3 test set. The training data $\mathcal{D}$ is LRS3 (S) or a combination of LRS3, VoxCeleb2 and AVSpeech (L). S contains 438 hours, L contains 3,068 hours, L^* contains 1,759 hours, $L^\dagger$ contains 3,448 hours and $L^\ddagger$ contains 100k private hours.*

Methods	θ_0	#Params	Sparsity	$\mathcal{D}$	WER (%)	Trainable
Small models						
Dense	Random	56 M	0	S	100.0	✗
LayerScale	Random	56 M	0	S	**42.8**	✓
MSRS	Random	56 M	0	S	48.5	✓
MSRS	Random	32 M	43.8	S	47.3	✓
Large models						
Dense	Random	250 M	0	S/L	100.0	✗
AVHubert [2]	Random	325 M	0	S	62.3	✗
RAVEn [3]	Random	493 M	0	S	87.3	✗
MSRS	Random	250 M	39.1	S	**46.9**	✓
AVHubert [2]	Encoder	325 M	0	L^*	26.9	✓
RAVEn [3]	Encoder	493 M	0	L^*	24.4	✓
SparseVSR [25]	Frontend	250 M	0	L	21.1	✓
Auto-AVSR [1]	Frontend	250 M	0	$L^\dagger$	20.5	✓
LP Conformer [30]	Encoder	0.57 B	0	$L^\ddagger$	**12.8**	✓
LayerScale	Random	250 M	0	L	21.9	✓
MSRS	Random	250 M	0	L	**21.1**	✓
MSRS	Random	151 M	39.7	L	23.9	✓
MSRS + LayerScale	Random	151 M	39.7	L	**23.2**	✓

4.2. Exploring the minimal amount of training data required for model convergence

To investigate the effect of data in this scenario, we trained our large VSR model with MSRS using varying amounts of training data and created sparse versions of it. As shown in Table 2, a sparse VSR model converges with an increase of up to a 5% in WER when the data reduction reaches 97%. This suggests the potential for training large models in situations where data is limited, such as for low resource languages. However, as the dataset continues to shrink, the pruning process slows down, which may require further adjustments to the λ parameter.

Table 2: *Sparse large VSR model trained with MSRS on varying amounts of data from a combination of the LRS3, VoxCeleb2, and AVSpeech datasets. Evaluation on the LRS3 test set.*

Data (%)	Data (hrs)	Epochs (j + T)	Sparsity	WER (%)	Trainable
100	3,068	3 + 75	39.7	23.9	✓
75	2,301	3 + 75	39.2	24.1	✓
50	1,534	3 + 75	39.2	24.2	✓
25	767	3 + 150	39.1	25.3	✓
10	307	5 + 300	39.1	26.7	✓
5	153	10 + 600	39.1	26.9	✓
3	92	15 + 800	39.2	28.5	✓
1	31	45 + 950	39.1	37.7	✓

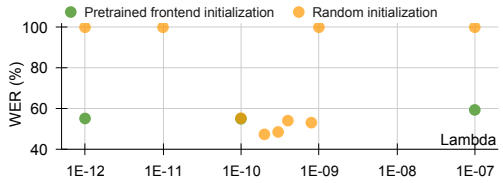

Figure 2: *Impact of λ on VSR trainings.*

4.3. Impact of the pruning speed on training from scratch

The success of training speech models from scratch using MSRS is highly dependent on the hyperparameter λ. An ablation study has been conducted to investigate the impact of different λ values on the small VSR setup trained on LRS3, as shown in Figure 2. The results indicate that when a good initialization (pretrained frontend) is used, λ values within the range of 1e-12 to 1e-7 effectively prune the model to a near-optimal sparsity of around 40%. Specifically, larger values of λ result in faster pruning but also make recovery more difficult and lead to accuracy drops, while smaller values do not force enough weights to be negative, which will eventually result in a denser model. In the context of training from scratch, it is crucial to progressively eliminate weights that hinder the training process. We achieve this by zero-outing those weights that negatively impact the loss until a near-optimal sparsity is attained and a healthier gradient flow is established. Therefore, the target range of λ becomes much smaller (1e-10 to 1e-9), with 2e-10 being the best trade-off in our all experiments. Note that if the dataset size decreases significantly below 300 hours (as shown in Table 2), it may be necessary to adjust the value of λ to maintain the pruning speed.

4.4. End-to-end training of AVSR models from scratch

Table 3 presents the AVSR results that illustrate the capability of other input-based speech models to be trained from scratch using MSRS. Even though, our proposed AVSR model with small capacity can be fully trained from scratch without additional regularization, it is crucial to use regularization as the model size increases to ensure convergence. Therefore, we explore multiple capabilities of MSRS and utilize a very large model (700 M parameters) that benefits from depth. Specifically, we leverage MSRS to require minimal data for convergence and accelerate training by using less than 1k hours of the available data (only 30%) for MSRS training with 16-float precision, and then transitioned to a dense model trained on the entire dataset to achieve SoTA results. Our very large model outperforms the current SoTA method presented in [1] without using any pre-training strategies under lower bit precision. This is particularly evident under very noisy conditions, where we observe more than a 3% WER absolute improvement. This is not the case for LayerScale, where a lower precision model does not converge.

4.5. Comparison with other sparse-based training methods

We compare various pruning methods for neural networks in Table 4, focusing on those that achieve the desired sparsity during or at the start of training. One method is GMP [16], which gradually prunes the smallest magnitude weights during training to reach the target sparsity. Another approach is sparse-to-sparse training, which starts with the target sparsity and uses a prune-and-grow approach to explore the parameter space. Notable methods in this category include SET [17], RigL [18] and CHASE [19], which use Erdos-Rényi-Kernel (ERK) [17, 18] to initialize models at the target sparsity and then prune weights based on magnitude or gradient. Our results show that when a pretrained frontend is given, all sparse strategies work without much degradation, but when the model is randomly initialized, only MSRS converges. The main difference between MSRS and other methods is that MSRS selectively prunes weights that negatively impact the loss, resulting in a narrower model that enables proper gradient flow. In contrast, other sparse methods employ initialization and pruning strategies that are decoupled from the training process. Figure 3 shows the sparsity for each module using different pruning methods. We observe that MSRS prunes frontend and encoder layers significantly to improve gradient flow, while ERK initialization primarily targets layers with more weights in the frontend, which results in less healthy flow. Our insight is that MSRS solves the vanishing gradient issue by properly reducing model complexity. This is achieved by cutting down small gradients, which are the root cause of the issue. To prove that, we applied the generated masks from MSRS training to the gradients of a dense model during training and found improved convergence (Fig. 1 - green line), with 52.1% WER, only 5% up. This implies that the masked weights are those that experience gradient vanishing.

Table 3: *WER (%) of AVSR models as a function of the noise levels on the LRS3 test set. Models are trained using a combination of LRS3, VoxCeleb2, and AVSpeech datasets in the presence of babble noise*. "Reg." refers to regularization.*

Reg. / SNR (dB)	#Params	-7.5	-2.5	2.5	7.5	12.5	Clean	Trainable
Auto-AVSR [1]	443 M	5.6	2.2	1.5	1.0	1.0	0.9	✓
Dense	268 M	2.8	1.5	1.1	1.0	0.9	**0.8**	✓
Dense MSRS	268 M	2.6	1.5	1.1	1.0	0.9	0.9	✓
Sparse MSRS (39.7%)	162 M	3.6	1.8	1.5	1.1	1.0	1.0	✓
Dense MSRS†	703 M	**2.5**	**1.3**	**1.0**	**0.9**	**0.9**	0.9	✓

*Training/testing data augmented with babble noise from NOISEX [31]. †Precision FP-16.

Table 4: *VSR models sparsified with different strategies on the LRS3 test set. Models follow different initializations θ_0.*

Methods	θ_0	#Params	Sparsity	WER (%)	Trainable
SparseVSR [25]	Model	34 M	40.0	38.5	✓
MSRS	Model	33 M	41.7	39.0	✓
SparseVSR [25]	Frontend	56 M	0	39.3	✓
SET [17]	Frontend	34 M	40.0	44.6	✓
RigL [18]	Frontend	34 M	40.0	44.4	✓
GMP [16]	Frontend	34 M	40.0	44.7	✓
CHASE [19]	Frontend	34 M	40.0	42.7	✓
MSRS	Frontend	33 M	41.7	**42.5**	✓
Dense	Random	56 M	0	100.0	✗
SET [17], RigL [18] GMP [16], CHASE [19]	Random	34 M	40.0	100.0	✗
MSRS	Random	32 M	43.8	**47.3**	✓
Dense with MSRS masks	Random	56 M	0	52.1	✓

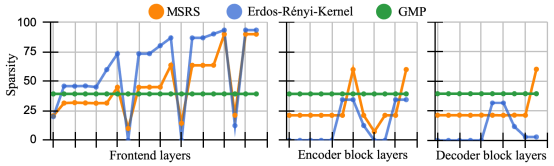

Figure 3: *Sparsity distribution in the frontend, an encoder block and a decoder block using various approaches: MSRS, ERK (for SET, RigL, and CHASE) and GMP. For encoder and decoder, the average/block is presented, respectively. Non-prunable layers like normalization layers are excluded.*

5. Conclusions

In this work, we introduce a sparse regularization approach that focuses on quickly developing sparse patterns within dense models to promote gradient flow, enabling end-to-end training of VSR/AVSR models from scratch. Our algorithm efficiently obtains a nearly optimal mask in just a few epochs and the models can be transferred back to dense or kept sparse until completion. Our experiments show competitive results in both VSR and AVSR modalities, with AVSR particularly standing out for its improvements in noisy scenarios. MSRS is the only sparse technique that addresses the convergence issues faced by dense models when training from scratch, by selectively masking the weights affected by vanishing gradients. Moreover, it allows lower bit precision without adding extra parameters.

6. Acknowledgements

Only non-Meta authors conducted any of the dataset pre-processing (no dataset pre-processing took place on Meta's servers or facilities). Shiwei Liu is supported by the Royal Society with the Newton International Fellowship.

7. References

[1] P. Ma, A. Haliassos, A. Fernandez-Lopez, H. Chen, S. Petridis, and M. Pantic, "Auto-avsr: Audio-visual speech recognition with automatic labels," in *Proceedings of ICASSP*, 2023, pp. 1–5.

[2] B. Shi, W.-N. Hsu, K. Lakhotia, and A. Mohamed, "Learning audio-visual speech representation by masked multimodal cluster prediction," in *Proceedings of ICLR*, 2022.

[3] A. Haliassos, P. Ma, R. Mira, S. Petridis, and M. Pantic, "Jointly learning visual and auditory speech representations from raw data," in *Proceedings of ICLR*, 2023.

[4] T. Afouras, J. S. Chung, A. Senior, O. Vinyals *et al.*, "Deep audio-visual speech recognition," *IEEE Transactions on PAMI*, 2018.

[5] J. Son Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," in *Proceedings of CVPR*, 2017, pp. 6447–6456.

[6] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of ICML*, 2009, pp. 41–48.

[7] P. Ma, B. Martinez, S. Petridis, and M. Pantic, "Towards practical lipreading with distilled and efficient models," in *Proceedings of ICASSP*, 2021, pp. 7608–7612.

[8] P. Ma, S. Petridis, and M. Pantic, "Visual speech recognition for multiple languages in the wild," *Nature Machine Intelligence*, vol. 4, no. 11, pp. 930–939, 2022.

[9] ——, "End-to-end audio-visual speech recognition with conformers," in *Proceedings of ICASSP*, 2021, pp. 7613–7617.

[10] K. Prajwal, T. Afouras, and A. Zisserman, "Sub-word level lip reading with visual attention," in *Proceedings of CVPR*, 2022, pp. 5162–5172.

[11] Y. A. D. Djilali, S. Narayan, E. LeBihan, H. Boussaid, E. Almazrouei, and M. Debbah, "Do vsr models generalize beyond lrs3?" in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 6635–6644.

[12] H. Zhang, Y. N. Dauphin, and T. Ma, "Fixup initialization: Residual learning without normalization," in *Proceedings of ICLR*, 2018.

[13] T. Bachlechner, B. P. Majumder, H. Mao, G. Cottrell, and J. McAuley, "Rezero is all you need: Fast convergence at large depth," in *Uncertainty in Artificial Intelligence*, 2021, pp. 1352–1361.

[14] X. S. Huang, F. Perez, J. Ba, and M. Volkovs, "Improving transformer optimization through better initialization," in *Proceedings of ICML*, 2020, pp. 4475–4483.

[15] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Proceedings of ICCV*, 2021, pp. 32–42.

[16] M. Zhu and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression," *Proceedings of ICLR*, 2018.

[17] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta, "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science," *Nature communications*, vol. 9, no. 1, p. 2383, 2018.

[18] U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen, "Rigging the lottery: Making all tickets winners," in *Proceedings of ICML*. PMLR, 2020, pp. 2943–2952.

[19] L. Yin, G. Li, M. Fang, L. Shen, T. Huang, Z. Wang, V. Menkovski, X. Ma, M. Pechenizkiy, S. Liu *et al.*, "Dynamic sparsity is channel-level sparsity learner," *Proceedings of NeurIPS*, vol. 36, 2024.

[20] Y. Gao, N. Colombo, and W. Wang, "Adapting by pruning: A case study on bert," *arXiv preprint arXiv:2105.03343*, 2021.

[21] N. Colombo and Y. Gao, "Differentiable architecture pruning for transfer learning," *arXiv preprint arXiv:2107.03375*, 2021.

[22] ——, "Disentangling neural architectures and weights: A case study in supervised classification," *arXiv preprint arXiv:2009.05346*, 2020.

[23] S. Liu, D. C. Mocanu, A. R. R. Matavalam, Y. Pei, and M. Pechenizkiy, "Sparse evolutionary deep learning with over one million artificial neurons on commodity hardware," *Neural Computing and Applications*, vol. 33, pp. 2589–2604, 2021.

[24] T. Afouras, J. S. Chung, and A. Zisserman, "LRS3-TED: a large-scale dataset for visual speech recognition," *Preprint at https://arxiv.org/abs/1809.00496*, 2018.

[25] A. Fernandez-Lopez, H. Chen, P. Ma, A. Haliassos, S. Petridis, and M. Pantic, "SparseVSR: Lightweight and noise robust visual speech recognition," *Proceesings of Interspeech*, 2023.

[26] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proceedings of Interspeech*, 2018, pp. 1086 – 1090.

[27] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation," *ACM Transactions on Graphics*, vol. 37, no. 4, p. 112, 2018.

[28] R. Errattahi, A. El Hannani, and H. Ouahmane, "Automatic speech recognition errors detection and correction: A review," *Procedia Computer Science*, vol. 128, pp. 32–37, 2018.

[29] S. Petridis, T. Stafylakis, P. Ma, F. Cai, G. Tzimiropoulos, and M. Pantic, "End-to-end audiovisual speech recognition," in *Proceedings of ICASSP*, 2018, pp. 6548–6552.

[30] O. Chang, H. Liao, D. Serdyuk, A. Shahy, and O. Siohan, "Conformer is all you need for visual speech recognition," in *Proceedings of ICASSP*, 2024, pp. 10 136–10 140.

[31] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech communication*, vol. 12, no. 3, pp. 247–251, 1993.