SAGA: Exploring Audio and Video Extensions for the nano4M Multimodal Framework

Sacha Godey (362191), Alexis Carreras (361573), Gabriel Taieb (360560), Adrien Bousquié (361516)
COM-304 Final Project Report

***Abstract*—This work addresses the challenge of extending the nano4M architecture to support dynamic audio and video modalities. Our approach leverages the VGGSound dataset [1] and adapts the nano4M framework, utilizing four NVIDIA V100 GPUs for training. We demonstrate that nano4M can be trained on video data, but the process is highly resource-intensive and constrained by the complexity of the modalities. Our experiments reveal that the masking strategy should be done with an homogeneity in the modality tokenization, which imposes additional constraints on training dynamics. Results for the audio modality were suboptimal, indicating a need for further investigation and improved audio representation. Overall, this study highlights both the potential and the current limitations of scaling nano4M to richer multimodal tasks.**

I. Introduction

In this project, we explore the extension of the Nano4M model to support dynamic modalities such as video and audio. Since 4M (and by extension Nano4M) has been created to interact effectively on static data like images and text, it lacks the ability to process time-based modalities, which limits its broader applicability. To address this, we experimented with a balanced approach: instead of directly processing raw video or audio signals, we extracted representative frames from videos in the VGG-Sound dataset [1] and trained the model on these static snapshots. This allowed us to remain compatible with the original 4M architecture, while still introducing elements of dynamic data. We also conducted initial experiments with hyperparameter tuning to improve generation quality. Our results are modest, but they help clarify some of the difficulties involved in extending Nano4M to dynamic modalities and offer initial directions for more exploration in the domain.

II. Related Work

Cross-modal generation has traditionally been addressed with specialised, task-specific systems. Recent diffusion systems such as Dreamix [2] convert a single image (plus text prompt) into a temporally-coherent clip, thus addressing the *image* $\rightarrow$ *video* setting. Conversely, models like Soundify [3] and Video2Sound [4] learn to generate plausible Foley tracks from silent footage, i.e. *video* $\rightarrow$ *audio*; however, each of these systems is trained for one fixed mapping and cannot perform fully flexible any-to-any transfer. AudioLDM [5] focuses on *text*$\rightarrow$*audio*, while Make-A-Video [6] tackles *text*$\rightarrow$*video*. Although these methods reach high perceptual quality, each solves a single mapping and requires hundreds of millions, often billions, of parameters.

Unified multimodal models. Flamingo [7] and Kosmos-2 [8] extend large language models with visual adapters, enabling a range of *vision*$\rightarrow$*text* tasks. The 4M framework [9] went further by demonstrating *any-to-any* transfer between static image modalities (RGB, depth, segmentation, text) via discrete tokens. However, 4M does not include *temporal* streams such as video or audio.

Tokenisation enablers. Recent neural codecs make it feasible to treat continuous media as tokens: Mimi [10] compresses waveforms into $\sim$1 k symbols, and Cosmos tokenisers [11] achieve similar rates for images and short videos. For the scope of the communication projet, other students have begun to experiment with these tools in miniature settings, but to date none has reported stable *audio–video any-to-any* generation.

Our position. We build on the 4M recipe but adopt Mimi and Cosmos to add audio and video streams, and keep the backbone small ($\sim$100 M parameters) to stay within the SCITAS cluster limitations. Our results are preliminary, synchronisation remains imperfect and quality lags behind large models, but they constitute an empirical probe into the limits of compact, student-scale architectures for fully general audio-visual generation. Negative findings (e.g. loss imbalance, memory ceilings) inform the community about open challenges and guide future work.

III. Methods

A. Overview and Approach

Our primary objective is to extend the nano4M architecture to support two additional modalities: audio and video. We approached this by leveraging pre-trained tokenizers for efficient and consistent feature extraction and by selecting a large-scale multimodal dataset suitable for both audio and video tasks. Design choices were guided by prior experience, computational constraints, and the need for reproducibility.

B. Dataset

We used the VGGSound dataset [1], which comprises over 200,000 YouTube video links with corresponding audio content and descriptive labels. This dataset was chosen for its scale and the richness of its multimodal content, particularly

for the audio task. Each sample in our adapted pipeline consists of:

- **Video:** 9 frames extracted per sample, with a temporal spacing of 0.07 seconds between frames (0.63 seconds clip).
- **Audio:** 5-second audio clips sampled at 24,000 Hz.

To further enrich the dataset for video-based tasks, we augmented each sample by generating a depth map of the middle video frame using Depth Anything V2 model [12]. This additional modality was intended to provide spatial context, although the sparsity and variability of video content in VGGSound posed challenges for consistent frame extraction and may limit the video generation results. The large size of the dataset also imposed computational and time constraints on our experiments.

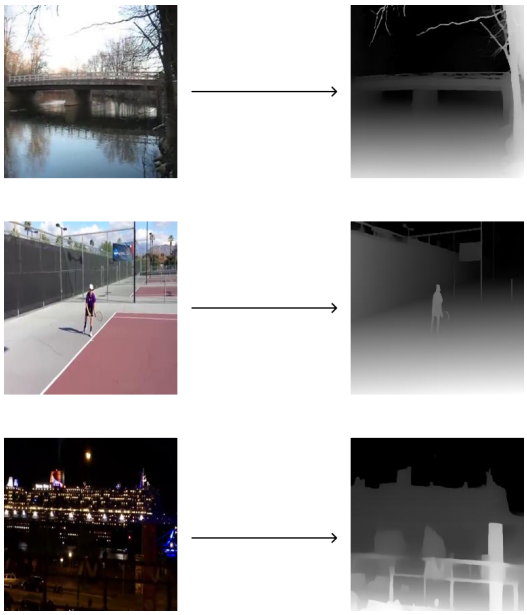

Figure 1. Example of VGGSound [1] content and corresponding depth map.

C. Tokenization

For modality-specific tokenization, we adopted the following pre-trained models:

- **Image and Video:** Cosmos tokenizers [11] were used due to their proven performance and compatibility with our existing pipelines.
- **Audio:** The Mimi tokenizer from the Moshi foundation model [10] was selected for audio tokenization, as it provided robust representations for our audio segments. (Note: The WavTokenizer was considered as an alternative for audio, but was ultimately not integrated in this iteration.)

All tokenizers were applied to the preprocessed dataset prior to model training, resulting in a unified, tokenized multimodal dataset.

D. Model Architecture

We based our experiments on the nano4M architecture, a compact adaptation of the 4M model [9]. The model comprises approximately 120 million parameters and is designed for efficient multimodal learning. The 4M (Massively Multimodal Masked Modeling) architecture is a unified Transformer encoder-decoder designed to process diverse modalities—such as images, audio, and video—by converting each into discrete tokens using modality-specific tokenizers [9]. All modalities are mapped into a shared token space, enabling full parameter sharing and seamless multimodal integration.

In both 4M and its compact nano4M variant, the encoder embeds input tokens with modality and positional information, while the decoder reconstructs masked tokens or predicts sequences using cross-attention. During training, a masked modeling objective is used: random subsets of tokens from all modalities are masked and predicted, which encourages the model to learn cross-modal relationships efficiently.

The nano4M model retains this architecture and fusion strategy but is scaled down (about 120 million parameters) for efficient experimentation in limited-resource settings.

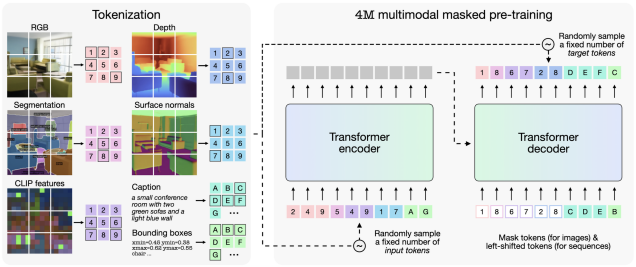

Figure 2. Overview of the 4M/nano4M architecture. [9]

E. Training and Infrastructure

Model training was conducted on the EPFL SCITAS Izar cluster, using four NVIDIA V100 GPUs. Training sessions lasted up to 9 hours per run. We used the WandB platform for experiment tracking, hyperparameter logging, and result visualization.

F. Procedure

1) **Data Collection and Preprocessing:**
   - Downloaded video and audio content from YouTube links in VGGSound.
   - Extracted video frames and audio clips according to defined modality constraints.
   - Generated depth maps for middle video frames.
2) **Tokenization:**
   - Applied Cosmos tokenizers to image and video data.
   - Applied Mimi tokenizer to audio data.

- Prepared a pre-tokenized dataset for efficient model training.

3) **Model Training:**
   - Trained the nano4M model on the prepared dataset, experimenting with various hyperparameters to optimize performance and diagnose potential issues.
4) **Evaluation:**
   - Evaluated the best-performing model configurations by adjusting inference parameters and analyzing output quality.

G. Design Choices and Alternatives

We prioritized pre-trained tokenizers for their robustness and ease of integration, given limited computational resources. While the VGGSound dataset was optimal for audio, its video content was less consistent, which may affect the effectiveness of video modality extensions. Future work could explore alternative datasets with denser and higher-quality video content or integrate additional tokenizers such as WavTokenizer for further audio analysis. WavTokenizer as actually been trained and prepared but we haven't had the time to use it for our model.

IV. Experiments

A. Baselines

We defined two internal baselines:

- **Joint Training (All Modalities Simultaneously):** Firstly, the model was trained on image, depth, video, and audio tokens in a unified setup with shared attention and output heads.
- **Simplified Modality Training:** On a second hand, we trained the model on a reduced subset of modalities (e.g., only image + depth or only image + audio) to evaluate whether a lower intermodal complexity improved convergence.

B. Joint Training Results and Challenges

When trained jointly on all modalities, the model failed to converge properly. The audio stream, due to its longer token sequence, disproportionately dominated the attention mechanism and loss function. This resulted in unstable gradients and poor reconstructions across all modalities. Generated video frames were mostly blank unless inference temperature was drastically increased, leading to oversaturated or noisy outputs.

Furthermore, the masking strategy proved problematic in joint setups: shorter modalities (e.g., image or depth) were often underrepresented in loss contributions, skewing training toward modalities with longer token sequences (like audio).

C. Simplified Training Outcomes

Reducing the number of active modalities resulted in a modest improvement in training stability. For instance, training the model on pairs such as image + depth or image + audio enabled it to learn more coherent cross-modal relationships. As part of our experiments, we also attempted conditioning the model on class labels to guide generation. However, the results remained inconclusive. One possible explanation is our simplistic approach to label representation—encoding each class as a single numeric token—which may have failed to provide sufficient semantic context. Although the generated outputs remained far from realistic, certain examples indicated that the model had begun to capture non-trivial associations across modalities.

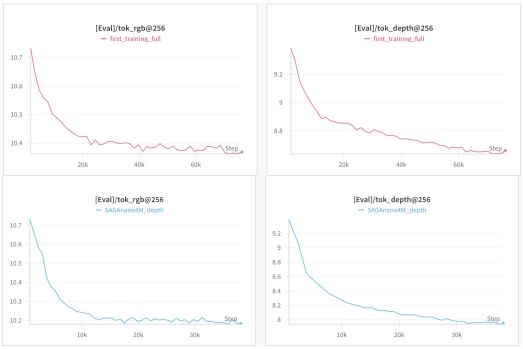

Figure 3. Comparison of losses. Up is the joint training, down is the simplified (RGB and Depth)

D. Inference and Temperature Tuning

Generation quality was highly sensitive to the inference temperature. In early trials, default temperatures produced empty or incoherent outputs. Raising the temperature to values between 0.6 and 0.8 was necessary to introduce diversity in the output tokens and produce interpretable reconstructions. This suggests that the model's token distribution remained overconfident due to under-training.

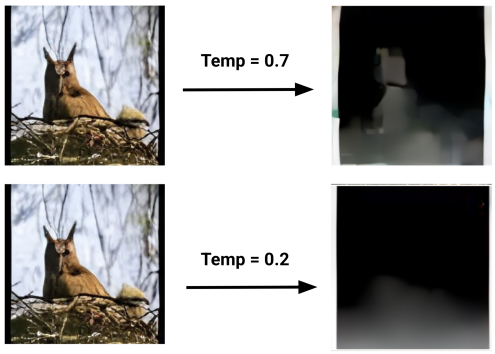

Figure 4. Exemple of generation with different values of temperature on the joint trained model

V. Conclusion and Limitations

In this project we set out to push *nano4M*, a lightweight variant of the 4M any-to-any multimodal model—beyond its original static-image scope by adding **tokenized video and audio**. Leveraging Cosmos (vision) and Mimi (audio) neural codecs, we built a unified, pre-tokenized version of the VGGSound dataset and trained a ~ 120 M-parameter encoder–decoder on four NVIDIA V100 GPUs. Despite tight resource limits, the system learned cross-modal correlations and produced recognisable, if low-fidelity, audio–visual generations. The work therefore establishes a student-scale baseline for fully token-based audio–video modelling and highlights the trade-offs involved in bringing temporal modalities into compact architectures.

A. Key Takeaways

- **Proof-of-concept extension.** We demonstrated that the nano4M recipe, originally designed for static images and text, can be trained on *tokenised* video and audio streams when supplied by modern neural codecs (Cosmos for vision, Mimi for sound).
- **Resource-aware design.** By capping the model at $\sim$120 M parameters and restricting training to four V100 GPUs, we mapped out the limits of "student-scale" hardware for genuinely multimodal modelling.
- **Modality interaction insights.** Stable learning requires (i) homogeneous token budgets across modalities and (ii) a masking ratio that does not systematically starve the shorter streams in favour of the longer ones.

B. Current Limitations and Possible Remedies

1) **Under-training.** Nine-hour runs leave the model far from convergence. Longer schedules or gradient-accumulation across more GPUs / A100s—are the most direct fix.
2) **Dataset sparsity.** VGGSound provides a too sparse video dataset. A way to fixe it would be to focus on a smaller part of it. Or find a new, more cocentrated datset.

Possible Future extensions. Possible future extensions include building our own audio and video tokenizer to realise this task. Super Resolution to jointly train on low-resolution images (e.g. 224x224 pixels = 14x14 tokens), and high-resolution images (e.g. 448x448 pixels = 28x28 tokens) would also be a possible addition we would like to add to the project.

VI. Individual contributions

S.G downloaded and maintained the dataset while A.B implemented the tokenization process and G.T implemented the possibility to save the tokens. A.C implemented the possibility to load and forward the tokens into the model.

A.C, S.G and A.B tried experimenting with different training pipeline to improve the model result.

S.G trained and evaluated the WavTokenizer.

A.C, G.T and A.B wrote the paper.

G.T developed the website.

References

[1] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, "VGGSound: A Large-scale Audio-Visual Dataset," Sep. 2020, arXiv:2004.14368 [cs]. [Online]. Available: http://arxiv.org/abs/2004.14368

[2] E. Molad, E. Horwitz, D. Valevski, A. R. Acha, Y. Matias, Y. Pritch, Y. Leviathan, and Y. Hoshen, "Dreamix: Video Diffusion Models are General Video Editors," Feb. 2023, arXiv:2302.01329 [cs]. [Online]. Available: http://arxiv.org/abs/2302.01329

[3] D. C.-E. Lin, A. Germanidis, C. Valenzuela, Y. Shi, and N. Martelaro, "Soundify: Matching Sound Effects to Video," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. San Francisco CA USA: ACM, Oct. 2023, pp. 1–13. [Online]. Available: https://dl.acm.org/doi/10.1145/3586183.3606823

[4] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg, "Visual to Sound: Generating Natural Sound for Videos in the Wild," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, Jun. 2018, pp. 3550–3558. [Online]. Available: https://ieeexplore.ieee.org/document/8578472/

[5] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-Audio Generation with Latent Diffusion Models," Sep. 2023, arXiv:2301.12503 [cs]. [Online]. Available: http://arxiv.org/abs/2301.12503

[6] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman, "Make-A-Video: Text-to-Video Generation without Text-Video Data," Sep. 2022, arXiv:2209.14792 [cs]. [Online]. Available: http://arxiv.org/abs/2209.14792

[7] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, "Flamingo: a Visual Language Model for Few-Shot Learning," Nov. 2022, arXiv:2204.14198 [cs]. [Online]. Available: http://arxiv.org/abs/2204.14198

[8] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei, "Kosmos-2: Grounding Multimodal Large Language Models to the World," Jul. 2023, arXiv:2306.14824 [cs]. [Online]. Available: http://arxiv.org/abs/2306.14824

[9] D. Mizrahi, R. Bachmann, O. F. Kar, T. Yeo, M. Gao, A. Dehghan, and A. Zamir, "4M: Massively Multimodal Masked Modeling," Dec. 2023, arXiv:2312.06647 [cs]. [Online]. Available: http://arxiv.org/abs/2312.06647

[10] A. Défossez, L. Mazaré, M. Orsini, A. Royer, P. Pérez, H. Jégou, E. Grave, and N. Zeghidour, "Moshi: a speech-text foundation model for real-time dialogue," Oct. 2024, arXiv:2410.00037 [eess]. [Online]. Available: http://arxiv.org/abs/2410.00037

[11] NVIDIA, N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, D. Dworakowski, J. Fan, M. Fenzi, F. Ferroni, S. Fidler, D. Fox, S. Ge, Y. Ge, J. Gu, S. Gururani, E. He, J. Huang, J. Huffman, P. Jannaty, J. Jin, S. W. Kim, G. Klár, G. Lam, S. Lan, L. Leal-Taixe, A. Li, Z. Li, C.-H. Lin, T.-Y. Lin, H. Ling, M.-Y. Liu, X. Liu, A. Luo, Q. Ma, H. Mao, K. Mo, A. Mousavian, S. Nah, S. Niverty, D. Page, D. Paschalidou, Z. Patel, L. Pavao, M. Ramezanali, F. Reda, X. Ren, V. R. N. Sabavat, E. Schmerling, S. Shi, B. Stefaniak, S. Tang, L. Tchapmi, P. Tredak, W.-C. Tseng, J. Varghese, H. Wang, H. Wang, H. Wang, T.-C. Wang, F. Wei, X. Wei, J. Z. Wu, J. Xu, W. Yang, L. Yen-Chen, X. Zeng, Y. Zeng, J. Zhang, Q. Zhang, Y. Zhang, Q. Zhao, and A. Zolkowski, "Cosmos World Foundation Model Platform for Physical AI," Mar. 2025, arXiv:2501.03575 [cs]. [Online]. Available: http://arxiv.org/abs/2501.03575

[12] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth Anything V2," Oct. 2024, arXiv:2406.09414 [cs]. [Online]. Available: http://arxiv.org/abs/2406.09414

VII. Appendix

Your main report should be **4 pages maximum**. You can add your supplementary evaluations (e.g. additional qualitative results, non-important long experiments, etc) and method details to the appendix section here which does not have a page limit. *Make sure that the main material is provided in the main report.*

Website link: saga-com-304.github.io