SAGA: Source Attribution of Generative AI Videos Supplementary Material

Rohit Kundu[1,2], Vishal Mohanty[2], Hao Xiong[3], Shan Jia[2], Athula Balachandran[2], Amit K. Roy-Chowdhury[1]
[1]University of California, Riverside, [2]YouTube (Google), [3]Google DeepMind
{rohit.kundu@email, amitrc@ece}.ucr.edu; {rohitkun, vishalmohanty, haoxg, shanjia, athula}@google.com

This supplementary document provides additional details and experiments to complement the main paper. We provide the details on how the multi-granular attribution levels have been broken down in Sec. S-1, the complete implementation details for reproducibility in Section S-2. In terms of additional experiments and analyses, we provide state-of-the-art comparison for the attribution tasks in Section S-3, the ablation on the foundation model backbone in Section S-4, ablation on the amount of source labeled data used for Stage-2 of SAGA in Sec. S-5 and the ablation on performance based on the different loss objectives in Sec. S-6. Additionally, we show the consistency of T-Sigs in Section S-7.

S-1. Attribution Levels

The DeMamba dataset [5] enables a rigorous, multi-tiered approach to source attribution for AI-generated videos by leveraging detailed generator metadata. Table S-1 summarizes the key properties of each included generator, such as task type, Stable Diffusion (SD) version, development team, model lineage, and training sources. This structured metadata forms the basis for defining four distinct attribution levels in our framework: TASK-L, SD-L, TEAM-L, and GEN-L.

By grouping generators according to these criteria, we can systematically evaluate attribution performance under varying degrees of granularity. For example, GEN-L attribution requires distinguishing between individual generator models, while TEAM-L attribution focuses on identifying the broader development group responsible for the model. This design allows us to probe not only fine-grained source identification but also coarser, more interpretable groupings that may be relevant for practical forensic or policy applications.

To ensure clear attribution boundaries and avoid ambiguous cases, generators labeled as "Mixed" or "Unknown" in critical metadata fields are excluded from training and evaluation. This exclusion maintains the integrity of each attribution split and facilitates reproducible, interpretable benchmarking. Our framework thus provides a comprehensive foundation for future research in AI-generated video provenance, supporting both fine-grained and high-level attribution tasks across a diverse landscape of generation technologies.

S-2. Implementation Details

Details of Input Videos: Our pipeline extracts frames from each video using a constant frame rate (FPS) of 8. This value is chosen to align with the typical FPS of most generators in DeMamba [5]. SAGA processes videos of a fixed context window ($L = 8$). For videos shorter than L frames, we apply padding to maintain consistent input size. Since all manipulations span the entire video, each segment inherits the video's original label $y_k \in \{0,...,n_c-1\}$, where n_c is the number of classes dependent on the attribution task as follows:

- BIN-L: $n_c = 2$
- TASK-L: $n_c = 3$
- SD-L: $n_c = 5$
- TEAM-L: $n_c = 15$
- GEN-L: $n_c = 20$

Details of the SAGA architecture: The foundational vision encoder used to encode the frame-level features before utilizing them in our video transformer architecture is the SigLIP [26] model, specifically the SigLIP-So400m/14 [1] version, which has been trained on internet-scale data, and provides domain-agnostic features. The input image to the SigLIP model is resized to $g_m = \mathbb{R}^{378\times378\times3}$. SigLIP breaks down images in 14×14 non-overlapping patches for tokenization and feature extraction, resulting in (according to the notations followed in Section 3.1 of the main paper) $l_t = 729$ and $d_t = 1152$. SAGA's video transformer architecture (θ) depth is set to a total of 6 encoder layers, of which 1 is the spatial encoder layer and 5 are temporal encoder layers, which was chosen to balance feature complexity and computational efficiency. We specifically have more temporal encoder layers than spatial, since SigLIP already encodes spatial information in its features, and temporal feature modeling is the focus of θ. We adopt sine-cosine positional encodings [20] in our video transformer to inject temporal order into the tokenized frame embeddings. For frame index m and feature dimension d_t, the encoding is computed as,

$$P_{(m,2r+1)} = \cos\left(\frac{m}{10000^{\frac{2r}{d_t}}}\right), \quad P_{(m,2r)} = \sin\left(\frac{m}{10000^{\frac{2r}{d_t}}}\right), \quad (1)$$

Table S-1. Overview of the video generators included in the DeMamba dataset [5]. This table is central to our multi-tier attribution framework, as it details the grouping logic used to construct the TASK-L, SD-L, TEAM-L, and GEN-L splits for all attribution experiments. Generators labeled as "Mixed" or "Unknown" are excluded from training to ensure clear and consistent attribution boundaries. This organization enables rigorous, interpretable evaluation across diverse attribution granularities and sets a reproducible standard for future work in source attribution.

Generator	Task	SD Version	Team	Model name	Trained from
ZeroScope [17]	T2V	*Unknown*	Personal: Spencer Sterling	Zeroscope_v2 XL	ModelScope, version 1.4.2
I2VGen-XL [22]	I2V	SD 2.1	Alibaba Group	VideoComposer	ModelScope, LDM, SD 2.1+3DUnet
SVD [3]	I2V	SD 2.1	Stability AI	Stable Video Diffusion	SD 2.1, LVD-F
VideoCrafter [6]	T2V	SD 2.1	Tencent AI Lab	Videocrafter2	VideoCrafter1 (SD 2.1) + ModelScopeT2V
Pika [9]	*Mixed*	*Unknown*	Pika	-	-
DynamiCrafter [24]	I2V	SD 2.1	Tencent AI Lab	DynamiCrafter	VideoCrafter1 + SD 2.1
SD [28]	*Mixed*	SD 1.5	Shanghai AI Lab	PIA	SD 1.5
SEINE [7]	I2V	SD 1.4	Shanghai AI Lab	Short-to-long (S2L) video diffusion model	SD 1.4, Lavie
Latte [12]	T2V	SD 1.4	Shanghai AI Lab	Latent Diffusion Transformer	SD 1.4
OpenSora [29]	T2V	*Unknown*	HPC-AI Tech	Transformer-based video diffusion	2D VAE for v1.0, Video Diffusion Transformer DiT
ModelScope [21]	T2V	SD 2.1	Alibaba Group	ModelScopeT2V	SD 2.1
MorphStudio [18]	T2V	*Unknown*	MorphStudio	-	-
MoonValley [13]	T2V	*Unknown*	MoonValley	-	-
HotShot [14]	T2V	SDXL	Hotshot Co.	Hotshot-XL	Stable Diffusion XL
Show_1 [27]	T2V	*Mixed* (DeepFloyd+ModelScope)	Show Lab, National University of Singapore	Both pixel and latent VDM	Pixel + latent VDM
Gen2 [16]	*Mixed*	*Unknown*	Runway	RunwayML Gen	SD
Crafter [4]	T2V	SD 2.1	Tencent AI Lab	VideoCrafter1	SD 2.1
Lavie [23]	T2V	SD 1.4	Shanghai AI Lab	Cascaded video latent diffusion models	SD 1.4
Sora [10]	T2V	*Unknown*	OpenAI	-	-

where, r indexes the feature dimension. These encodings, added to the input tokens, provide temporal context for the transformer's attention mechanism with minimal computation cost.

Hardware, Software & Hyperparameters: SAGA is implemented on TensorFlow, and trained using an AdamW optimizer [11] (initial learning rate of $1e-05$, decay rate of 0.5 every 1000 steps), with a batch size of 64, trained for 10 epochs on 8 TPUv3 chips. The loss weight factor (Sec. 3.2 of the main paper) is set as $\lambda = 0.5$ and the margin hyperparameter α for both semi-HNM and HNM for our experiments was set as 1.0 according to popular choice in the literature. TensorFlow Data Service was used for multi-thread I/O during model training since the amount of video data (total $\sim$2M samples) is huge, and would otherwise reduce the duty cycle of the TPU. Stage-1 real/fake pretraining using all DeMamba [5] generators (80% data in training) takes $\sim$ 10hrs and Stage-2 attribution with 0.5% labeled data (which equates to 500 samples per class) takes $\sim$20mins. The end-to-end inference time per video on a single CPU takes $\sim$ 1min. The real/fake pretrained model (smallest SAGA version) has 95.64M trainable parameters, requiring 364.85 MB storage space, and the GEN-L trained model (largest SAGA) version has 95.66M trainable parameters requiring 364.93 MB storage space. Average TPU duty cycle (ratio of the amount of time TPU is actually performing calculations versus the amount of time for where TPU is booked and online) of the experiments was $\sim$80%.

S-3. SOTA Comparison on Attribution Tasks

We compare SAGA with both image-based and video-based state-of-the-art detection methods on all attribution granularities, as shown in Table S-2. SAGA consistently achieves much higher accuracy across all tasks, clearly outperforming existing approaches. These gains indicate that methods originally designed for binary deepfake detection or coarse video-level classification do not readily extend to fine-grained, multi-granular source attribution. In contrast, SAGA is explicitly tailored for video source attribution, and its architectural design and training strategy enable it to capture generator- and team-specific signatures, leading to superior performance across all levels.

Table S-2. SAGA vs. SOTA methods on the attribution tasks.

Method	Type	TASK-L	TEAM-L	SD-L	GEN-L
F3Net [15]	Image-based	60.76%	54.28%	48.32%	43.77%
NPR [19]	Image-based	67.35%	61.64%	52.19%	49.01%
TALL [25]	Video-based	84.92%	75.47%	72.61%	69.53%
SAGA	Video-based	**98.20%**	**97.77%**	**98.49%**	**94.99%**

Table S-3. Ablation of the foundation model backbone used as the frame-level vision encoder in SAGA.

Backbone	BIN-L	TASK-L	TEAM-L	SD-L	GEN-L
SigLIP-So400m/14	99.94%	98.20%	97.77%	98.49%	94.99%
DINOv2 ViT-L/14	99.56%	98.34%	97.69%	97.96%	94.63%

S-4. Ablation on Vision Encoder Backbone

We compare two foundation model backbones, SigLIP-So400m/14 (default) and DINOv2 ViT-L/14, as the frame-level vision encoder in SAGA, and report the results in Table S-3. Across all five evaluation granularities, the performance differences between the two backbones are marginal ($\leq$0.4%). This indicates that SAGA is remarkably robust to the specific choice of vision encoder and does not rely on a particular

pre-trained backbone to achieve strong attribution performance. Instead, the consistently high scores with both SigLIP and DINOv2 suggest that our gains primarily stem from the architectural design of SAGA and the proposed training strategy, rather than from backbone-specific advantages.

S-5. Ablation of Number of Labeled Samples

We observe that the performance of SAGA in Stage-2 contrastive finetuning is tied to both the number of labeled samples per class and the granularity of the attribution task as shown in Fig. S-1. As the number of labeled samples increases, accuracy improves across all attribution levels; however, the gains are most pronounced for fine-grained tasks such as team-level and generator-level attribution. Notably, as the number of classes increases with task granularity, the performance gap between low-data and high-data regimes widens substantially. This trend reflects the greater challenge of learning discriminative representations for a larger set of visually similar classes with limited supervision. These findings are consistent with prior work in contrastive learning, which highlights the importance of sufficient positive and negative samples for robust representation learning, especially in long-tailed or fine-grained classification settings [2, 8].

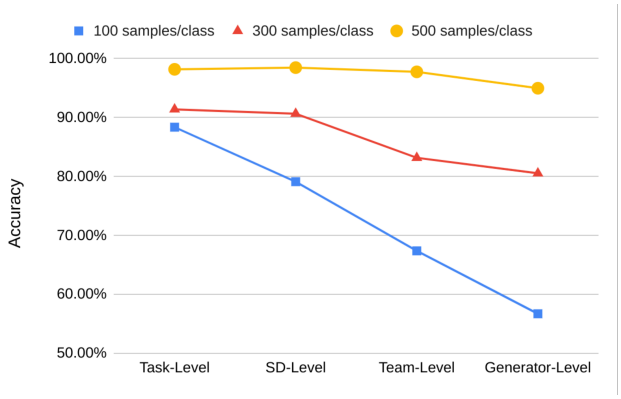

Figure S-1. Performance of SAGA as a function of labeled samples per class used in Stage-2 contrastive finetuning. Accuracy is shown for each attribution granularity. Results indicate that while high accuracy is maintained at coarser levels even with limited data, fine-grained attribution (team and generator-level) benefits significantly from increased labeled samples.

S-6. Ablation of Loss Functions

In Fig. S-2, we present a generator-wise comparison of SAGA under three loss configurations for the most challenging GEN-L attribution task: (1) cross-entropy (CE) only, (2) CE with semi-hard negative mining (semi-HNM), and (3) CE with hard negative mining (HNM). The HNM-augmented objective delivers uniformly strong accuracy across nearly all of the 19 generators, indicating that explicitly mining the closest negatives effectively enforces inter-generator separation even when classes are highly similar. By contrast, CE-only and semi-HNM show sporadic success, matching HNM on a small subset of generators, but frequently collapsing near 0.00% on many others, revealing insufficient separation of closely related generators. Averaged over classes, CE achieves 55.13%, semi-HNM improves to 70.13%, and HNM attains 94.99%, highlighting the substantial and consistent gains from hard-negative mining in this setting.

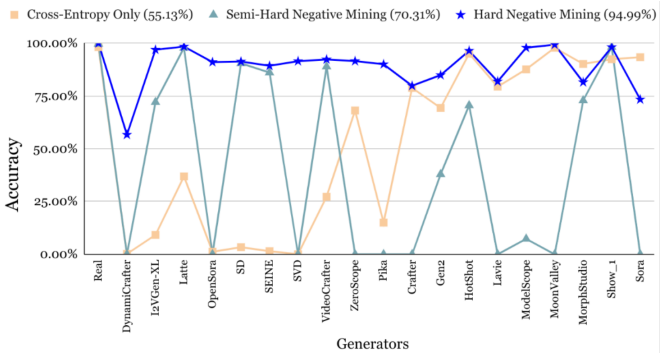

Figure S-2. Effect of loss function on GEN-L attribution performance.

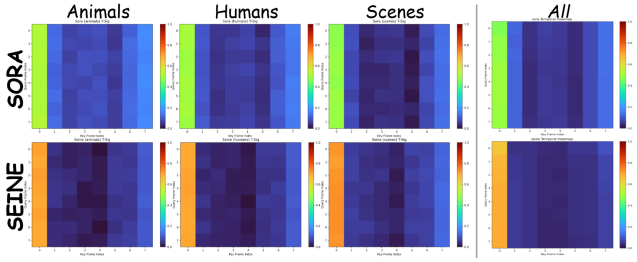

Figure S-3. T-Sigs stratified by video semantics. T-Sigs produce stable, consistent signatures regardless of video content.

These results are consistent with the margin-based view: semi-HNM prioritizes negatives within the margin but still farther than the positive, which can miss truly overlapping generators, whereas HNM targets the closest negatives that most directly violate or threaten the margin, leading to better cluster disentanglement.

S-7. T-Sig Consistency

To assess the robustness of T-Sigs to semantic content, we stratify videos by high-level categories (scenes, animals, humans) and visualize the corresponding signatures in Fig. S-3. Across these diverse content types, T-Sigs for a given generator exhibit consistent, model-specific patterns, indicating that the learned temporal traces are not tied to particular semantics. This supports our claim that T-Sigs capture generator-dependent temporal statistics that persist across different video contents, rather than spurious correlations with specific scene or object categories.

References

[1] Ibrahim M Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. Getting vit in shape: Scaling laws for compute-optimal model design. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[2] Alexandre Audibert, Aurélien Gauffre, and Massih-Reza Amini. Exploring contrastive learning for long-tailed multi-label text classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 245–261. Springer, 2024. 3

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[4] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 2

[5] Haoxing Chen, Yan Hong, Zizheng Huang, Zhuoer Xu, Zhangxuan Gu, Yaohui Li, Jun Lan, Huijia Zhu, Jianfu Zhang, Weiqiang Wang, et al. Demamba: Ai-generated video detection on million-scale genvideo benchmark. *arXiv preprint arXiv:2405.19707*, 2024. 1, 2

[6] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024. 2

[7] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. In *The Twelfth International Conference on Learning Representations*, 2023. 2

[8] Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*, 2020. 3

[9] Pika Labs. Pika art. https://pika.art/, 2022. 2

[10] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024. 2

[11] I Loshchilov and F Hutter. Decoupled weight decay regularization. *ICLR*, 2019. 2

[12] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024. 2

[13] moonvalley.ai. moonvalley.ai. https://moonvalley.ai/, 2022. 2

[14] John Mullan, Duncan Crawbuck, and Aakash Sastry. Hotshot-xl. https://github.com/hotshotco/hotshot-xl, 2023. 2

[15] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*, pages 86–103. Springer, 2020. 2

[16] Runway Research. Text driven video generation. https://research.runwayml.com/gen2, 2023. 2

[17] Spencer Sterling. Zeroscope-v2-xl. https://huggingface.co/cerspense/zeroscope_v2_XL, 2024. 2

[18] Morph Studio. Morph studio. https://www.morphstudio.com/, 2024. 2

[19] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28130–28139, 2024. 2

[20] A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A Gomez, Ł Kaiser, and I Polosukhin. Attention is all you need. *NeurIPS*, 2017. 1

[21] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 2

[22] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. I2vgen-xl. https://modelscope.cn/models/damo/Image-to-Video/summary, 2023. 2

[23] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023. 2

[24] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pages 399–417. Springer, 2024. 2

[25] Yuting Xu, Jian Liang, Gengyun Jia, Ziming Yang, Yanhao Zhang, and Ran He. Tall: Thumbnail layout for deepfake video detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22658–22668, 2023. 2

[26] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 1

[27] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *International Journal of Computer Vision*, pages 1–15, 2024. 2

[28] Yiming Zhang, Zhening Xing, Yanhong Zeng, Youqing Fang, and Kai Chen. Pia: Your personalized image animator via plug-and-play modules in text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7747–7756, 2024. 2

[29] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024. 2