AbsText2Video: Embracing Abstract Annotations to Caption Video Dataset

Fan Xie[1, 2], **Dan Zeng**[2, *], **Qiaomu Shen**[3], **Bo Tang**[1]

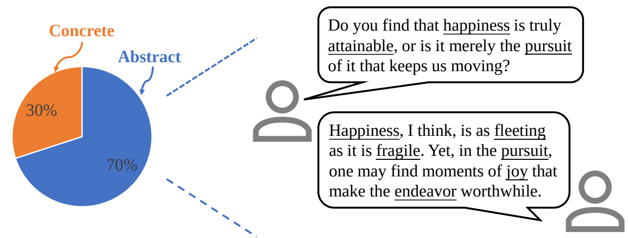
[1]Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
[2]School of Artificial Intelligence, Sun Yat-sen University, Zhuhai, China
[3]Beijing Institute of Technology (Zhuhai), Zhuhai, China
12232387@mail.sustech.edu.cn, zengd8@mail.sysu.edu.cn, joyshen06@gmail.com, tangb3@sustech.edu.cn

Abstract

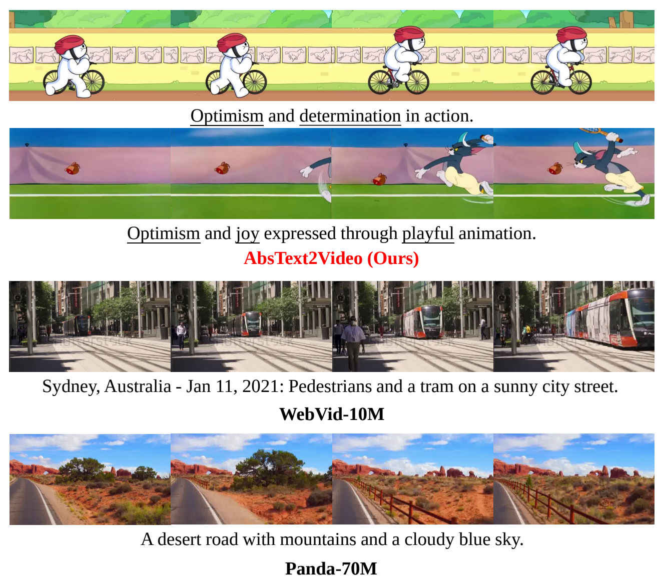
While text-to-video generation (T2V) methods have achieved astonishing success thanks to the advancement in large-scale T2V datasets, they suffer from a sharp performance drop on abstract description input. On the one hand, this is due to the lack of abstract text-to-video pairs in existing training data. On the other hand, it also stems from the ill-posed nature of the abstract text. There are many possible concrete texts corresponding to the same abstract text. More importantly, abstract language occupies a large proportion (over 70%) of our daily communication. To address this issue, we propose an LLM-based abstract text annotation pipeline that dynamically updates prompts based on the generation quality. In addition, we also propose the cycle similarity metric to measure the similarity between concrete and abstract text pairs. Finally, we introduce a new *AbsText2Video* dataset to push the video generation to a broader range of applications. Experiments on 11 T2V models verify the effectiveness of our dataset in tackling the abstract texts.

1 Introduction

Artificial Intelligence Generated Content (AIGC) has ushered in an unprecedented boom in the generation of multiple modalities such as text, image, audio, and video. To name a few, OpenAI's GPT-4(OpenAI et al. 2024) in text generation, Midjourney in image generation, Google's AudioLM(Borsos et al. 2023) in audio generation, and OpenAI's Sora(Brooks et al. 2024) in video generation. However, compared with other modalities, video generation lags. This stems from two reasons. First, it requires high temporal and spatial consistency, which is difficult to learn and becomes more challenging in the case of long video generation. Second, the existing T2V datasets are usually annotated with concrete, detailed descriptions, which inevitably leads to a performance drop in practical use as abstract description occupies around 70% of daily communication(Borghi et al. 2023). This discrepancy hinders models from accurately understanding and generating high-quality video. To the best of our knowledge, this paper is the first to introduce abstract text to video task, aiming to push the frontier of T2V generation from a dataset perspective.



(a) The proportion of word types and a conversation example.



(b) Examples for *AbsText2Video* and popular text-to-video datasets.

Figure 1: Embracing abstract text annotation is necessary to push frontier of text-to-video generation. Abstract words are underlined.

As illustrated in Fig. 1(a), daily communication involves both concrete and abstract descriptions, highlighting our ability as humans to engage in abstract thinking. Abstract words are underlined. Therefore, it is significant to explore generating video given abstract text, which is crucial to push the development of T2V to a broad range of applications. However, existing T2V datasets such as WebVid-10M(Bain

*Dan Zeng is the corresponding author.
Presented at the Workshop on Preparing Good Data for Generative AI: Challenges and Approaches (Good-Data) in conjunction with AAAI 2025.

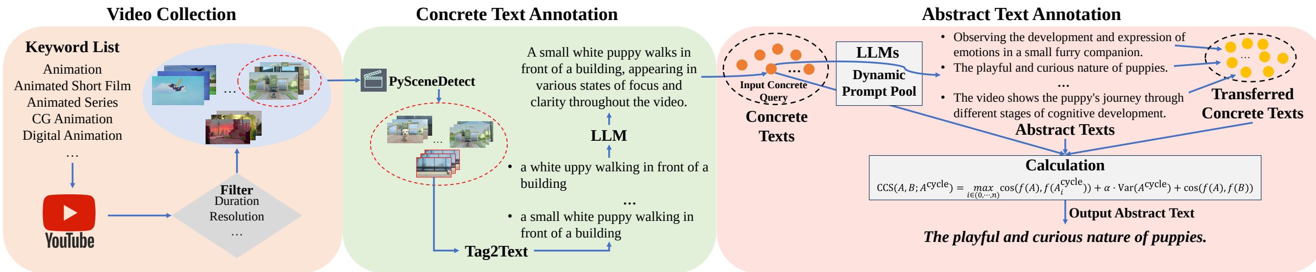


Figure 2: The proposed pipeline for annotating videos with abstract texts.

et al. 2021) and Panda-70M(Chen et al. 2024b) containing only concrete caption, which hinder the learning of abstract conception for video generation. To address this issue, we introduce a video dataset with abstract annotations, termed *AbsText2Video*, to foster development in generating videos from abstract texts. An example is shown in Fig. 1(b).

Nevertheless, annotating a video with an abstract caption given the concrete description is challenging for the following reasons. First, unlike the finite explanation of concrete descriptions, the abstract text has an ill-posed nature, leading to many possible concrete explanations corresponding to the same abstract text. Second, leveraging large language models (LLMs), a common approach for annotating concrete text, cannot be directly used to annotate abstract text because different LLMs may generate varying abstract captions for the same concrete text input. As the old saying goes, "There are a thousand Hamlets in a thousand people's eyes." Third, existing metrics such as SimCSE(Gao, Yao, and Chen 2021) and MiCSE(Klein and Nabi 2023) are used to evaluate concrete text, which are vulnerable to measuring the quality of abstract annotation because the embedding space is learned through concrete concepts during the training. A fair metric is needed to measure the quality of abstract annotations.

To overcome these challenges, we propose an abstract text annotation pipeline that leverages LLMs and in-context learning to dynamically update prompts based on the generation quality. In addition, we also propose the cycle similarity metric, **Cycle Consistency Similarity (CCS)**, to measure the similarity between concrete and abstract text pairs. It is based on the assumption that good abstract-concrete text pairs should be close to each other in the concrete embedding space if the abstract text is converted to concrete text. Finally, we introduce a new *AbsText2Video* dataset to push the video generation to a broader range of applications. Our dataset includes 200K abstract text-video pairs in two versions: 100K pairs sourced from the WebVid (Bain et al. 2021) dataset and another 100K pairs collected from YouTube.

2 Methodology

2.1 Abstract Text Annotation Pipeline Overview

As illustrated in Fig. 2, our abstract text annotation pipeline consists of video collection, concrete text annotation, and abstract text annotation. For video collection, we created a list of keywords of YouTube searches to filter videos with resolutions between 360p and 720p and durations between 10 seconds and 30 minutes. We break down text annotation into generating concrete text first and then annotating abstract text annotation based on concrete text because it is difficult for LLM to generate good abstract text. For concrete text annotation, we follow the conventional steps (e.g., Tag2Text, LLM) to build text-to-video dataset(Wang et al. 2023b,d). Once we obtain concrete text, we proceed to convert the abstract text.

Abstract Text Annotation Given an input concrete query, we utilize an LLM to generate transformed concrete texts based on our dynamic prompt pool. To ensure high annotation quality, we apply multiple LLMs, such as QWen(Bai et al. 2023), InternLM, and InternLM-8K(Cai et al. 2024). However, different LLMs often produce significantly varied abstract texts. To address this, we cycle back to N (e.g., $N = 10$) concrete texts based on the transformed abstract text. As a result, we obtain N triplet pairs – original concrete text, abstract text, and cycled concrete text pairs. We then calculate a similarity score for each pair using our CCS score. Only the concrete-abstract pair that yields the highest CCS scores will be selected for the final annotation. Dynamic prompt update strategy and the cycle consistency similarity score are detailed in the following sections.

2.2 Dynamic Prompt Update Strategy

Since good prompts are crucial for large language models to obtain high-quality abstract texts, we propose a dynamic updating strategy to iteratively update the demonstration examples in the prompts. Prior studies(Zhao et al. 2021; Liu et al. 2021; Sorensen et al. 2022; Gonen et al. 2022; Levy, Bogin, and Berant 2022; Lu et al. 2021) suggest that task-specific, high-quality examples can improve the performance of LLMs in conversion tasks. In light of this, our demonstration example consists of two parts: (1) predefined concrete-abstract pairs and (2) concrete-abstract pairs obtained based on similarity search, which is performed by searching for the K most similar pairs from the evaluation set given a query concrete text. The core idea is that if the new prompt (i.e., including the new pair) can produce a higher CCS score on the evaluation set than the old prompt, the new pair is included in the prompts. Meanwhile, the least accurate demonstration pairs in the prompts are removed ac-

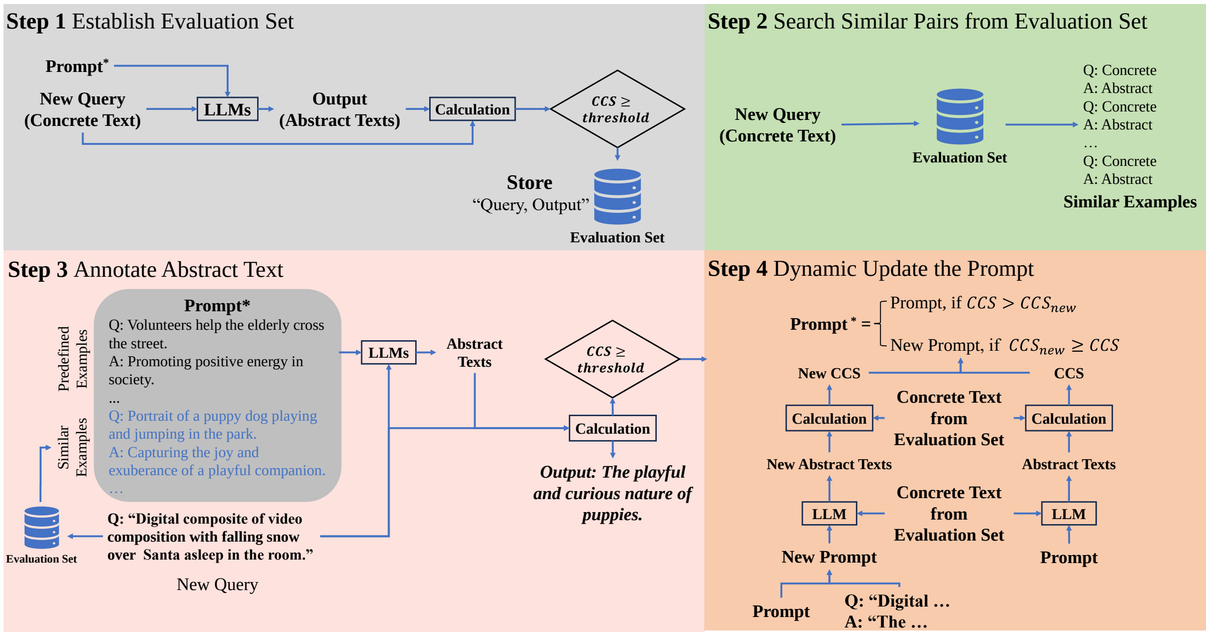


Figure 3: Illustrating the dynamic prompt update process.

cordingly.

As illustrated in Fig. 3, updating prompts dynamically consists of four steps. To begin, we establish the evaluation set by utilizing Prompt*. A large number of concrete texts are converted, and high-scoring examples are stored in the evaluation set. Once the evaluation set is established, the process of converting concrete texts begins. For each text to be converted, similar examples are identified from the evaluation set (Step 2). These similar examples are then combined with predefined examples to form Prompt*, which is subsequently used for text conversion (Step 3). If the CCS score of the generated abstract text exceeds the predefined threshold, the process proceeds to Step 4. In Step 4, the newly generated high-scoring example is added to the predefined examples to create a new prompt. This updated prompt and Prompt* are used to convert the texts in the evaluation set, after which the average CCS score is calculated. The prompt with the higher score is ultimately selected as the final Prompt*.

Due to computational constraints, the number of demonstration examples cannot increase indefinitely. After converting M texts or reaching the maximum allowable number of examples, we re-evaluate the examples in the evaluation set, removing the one whose exclusion leads to improved prompt performance.

2.3 Cycle Consistency Similarity Score

Given an input concrete text, we first generate an abstract text and then cycle back to generate N transformed concrete text. The proposed Cycle Consistency Similarity (CCS) in Equation 1 takes into account three factors: 1) the similarity between original and transformed concrete text, 2) the diversity of the generated N transformed concrete text in terms of embedding features, 3) the similarity between the concrete and abstract pair. In this way, we can expect a rather accurate abstract text from two perspectives. First, an accurate abstract text should be able to transform back to concrete text that is close to the original text. Second, unlike concrete text, abstract text has an ill-posed nature, and we should allow certain variations among several transformed concrete texts.

$$\mathrm{CCS}(A, B; A^{\mathrm{cycle}}) = \max_{i \in (0,\cdots,n)} \cos\left(f(A), f\left(A_i^{\mathrm{cycle}}\right)\right) + \alpha \cdot \mathrm{Var}(A^{\mathrm{cycle}}) + \cos\left(f(A), f(B)\right) \quad (1)$$

In Equation 1,A denotes the input concrete text, B represents the output abstract text, and A_i^{cycle} refers to the concrete text generated cyclically from B. The embeddings of A and B, denoted as $f(A)$ and $f(B)$, are obtained using the encoding method proposed by (Gao, Yao, and Chen 2021). The scaling factor α, set to 100 in this paper, ensures that the variance scale aligns with the similarity measure. The variance of A^{cycle} is defined as $\mathrm{Var}\left(A^{cycle}\right) = \frac{1}{N}\sum_{i=0}^{N-1}\left(f\left(A_i^{(cycle)}\right) - \frac{1}{N}\sum_{i=0}^{N-1} f\left(A_i^{(cycle)}\right)\right)^2$. An example of the calculation process is shown in Fig. 4.

3 Experiments

Benchmark dataset The *AbsText2Video* test set contains 10,000 videos with abstract captions generated using our method. 4,999 videos are from the validation set of the WebVid dataset, and the rest are from our collection. To ensure

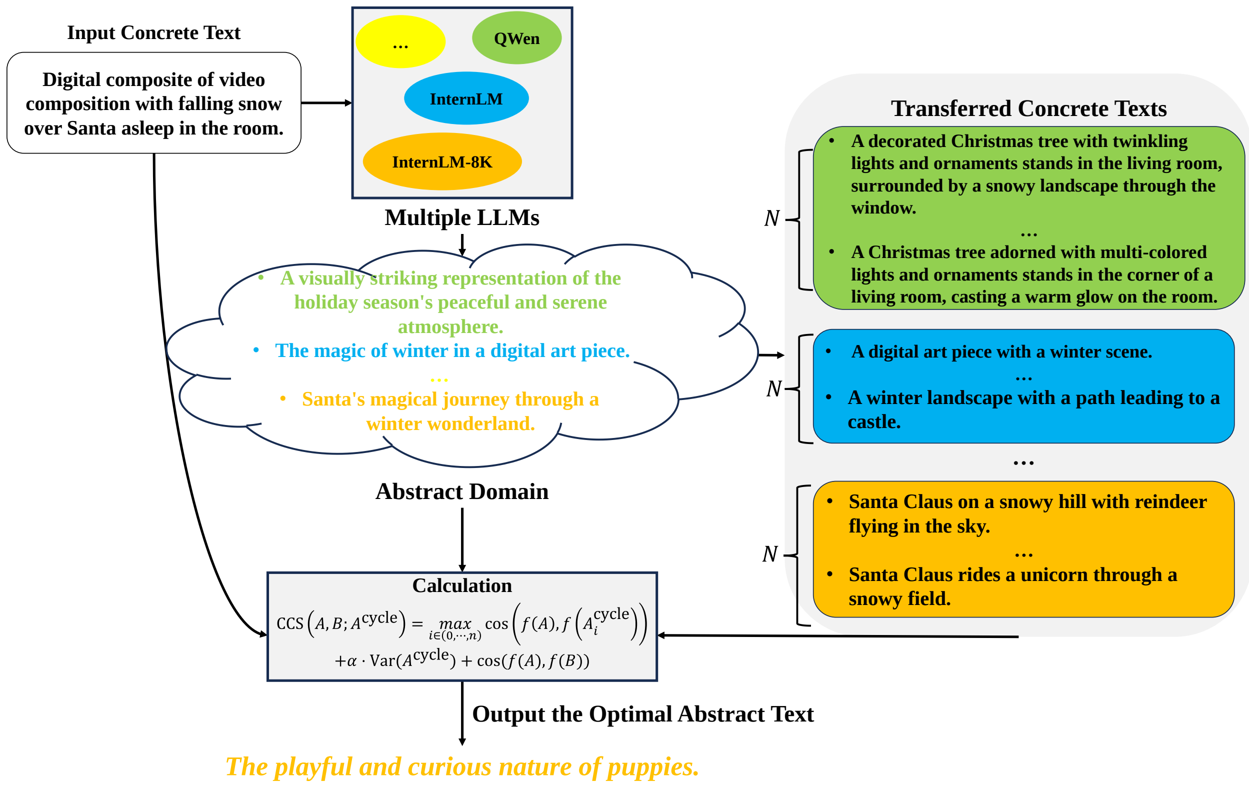


Figure 4: Example of CCS score calcultation.

a fair comparison, we standardize the output by instructing all models to generate 16-frame videos with a resolution of 256×256.

Evaluation Metric To assess video quality, we compute the Fréchet Video Distance (FVD) (Unterthiner et al. 2018) and the Video Quality Assessment (VQA) score (Wu et al. 2023). For evaluating the alignment between videos and abstract texts, we calculate the CLIP similarity (CLIPSim) (Wu et al. 2021). Specifically, CLIPSIM_1 measures the similarity between the video generated from the abstract text and the abstract text itself, CLIPSIM_2 measures the similarity between the video generated from the abstract text and the concrete text, and CLIPSIM_3 evaluates the similarity between the video generated from the concrete text and the concrete text.

3.1 Inference Performance Comparison

We select 11 popular generation models to generate videos based on the given abstract texts. The quantitative results are presented in Table 1. We can find that CLIPSIM_2 of all models is lower than CLIPSIM_1. This suggests that there is no embedding space that can encode both concrete and abstract text well. In addition, it is clear that CLIPSIM_3 of all models is higher than CLIPSIM_1, indicating that abstract text to video generation is indeed a difficult task even for the most state of the art T2V model.

3.2 LoRA Fine-tuning

We use the ModelScope model as an example, fine-tuning it with *AbsText2Video* to explore the impact of fine-tuning on the model's ability to generate videos from abstract texts. The fine-tuning is conducted using abstract text-video pairs from both WebVid and YouTube, which were independently collected for this study. The video quality in the YouTube subset outperforms that of the WebVid subset. The results are presented in Table 2. Because our *AbsText2Video* is significantly different from videos trained for baseline, the FVD of the fine-tuned version is worse compared to the baseline. Using *AbsText2Video* for finetuning does lead to better video generation quality in terms of VQA. As expected, CLIPSIM_1 between abstract text and generate video is also better after finetuning.

3.3 Effect of LLM Versions

We built upon VGen(Qing et al. 2024), utilizing *AbsText2Video* to train the model from scratch, subsequently using it to generate videos corresponding to abstract texts. Experiments were conducted on the YouTube subset and the YouTube Subset with new abstract annotations using the latest QWen2-VL model(Wang et al. 2024). As shown in Table 3, better LLM versions can lead to higher video quality in terms of all metrics, indicating that abstract annotation can be further improved if there are more advanced LLM.

Table 1: Performance results for abstract text of the current mainstream open source text-to-video models.

Year	Model	FVD ↓	VQA ↑	CLIPSIM_1 ↑	CLIPSIM_2 ↑	CLIPSIM_3 ↑
2022	LVDM(He et al. 2023)	850.18	28.16	0.2724	0.2436	0.3044
2023	ModelScope(Wang et al. 2023a)	531.20	25.63	0.2684	0.2361	0.3110
2023	VidRD(Gu et al. 2023)	**342.51**	29.41	0.2745	0.2349	0.2536
2023	LaVie(Wang et al. 2023c)	786.06	21.75	0.2597	0.2375	0.3031
2023	Show-1(Zhang et al. 2023)	1678.85	22.16	0.2752	0.2569	0.3089
2023	HotShot-XL(et al. 2023)	1561.45	32.76	0.2259	0.2083	0.2646
2023	FreeNoise(Qiu et al. 2024)	822.40	59.92	0.2765	0.2350	**0.3133**
2024	Latte(Ma et al. 2024)	219.91	61.86	0.2717	0.2376	0.3045
2024	VideoCrafter2(Chen et al. 2024a)	3897.06	34.31	**0.2812**	0.2429	0.3044
2024	Open-Sora(Zheng et al. 2024)	451.60	38.24	0.2673	0.2489	0.3002
2024	CogVideox(Yang et al. 2024)	1659.74	**68.77**	0.2699	0.2412	0.3046

Table 2: The quantitative results of ModelScope (1.7B) fine-tuned on *AbsText2Video* for generating videos from abstract texts.

Finetune Set	FVD ↓	VQA ↑	CLIPSIM_1 ↑
WebVid Subset	4369.23	21.88	0.2726
YouTube Subset	4402.86	**21.93**	**0.2733**
Baseline	**628.33**	21.32	0.1988

Table 3: Test results for training from scratch using *AbsText2Video* based on VGen.

Setting	FVD ↓	VQA ↑	CLIPSIM_1 ↑
YouTube Subset	1257.78	23.36	0.2353
YouTube Subset (New)	**1218.77**	**23.79**	**0.2398**

3.4 Ablation Study

We conducted an ablation study to examine the entire process of generating abstract text annotations for videos. The experiments were divided into multi-LLMs and the Dynamic Prompt Update Strategy, with the results presented in Table 4, which clearly indicate the effectiveness of our design for annotating video with abstract text.

4 Conclusion

To the best of our knowledge, this paper is the first to introduce abstract text to video task and build a new *AbsText2Video* dataset to push the video generation to a broader range of applications. We propose an LLMs-based abstract annotation pipeline with dynamic prompt update strategy and cycle consistency similarity (CCS) score. Extensive experiments on existing T2V models verify the effectiveness of our annotation pipeline and the importance of addressing abstract text to video generation.

Table 4: Ablation studies on annotation methods.

Multi-LLMs	Dynamic Prompt	CCS ↑	CLIPSIM_1 ↑
		1.63	0.2049
✓		2.34	0.2197
	✓	2.17	0.2216
✓	✓	**2.52**	**0.2284**

5 Acknowledgments

This work was supported by the National Natural Science Foundation of China No.62206123.

References

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; Hui, B.; Ji, L.; Li, M.; Lin, J.; Lin, R.; Liu, D.; Liu, G.; Lu, C.; Lu, K.; Ma, J.; Men, R.; Ren, X.; Ren, X.; Tan, C.; Tan, S.; Tu, J.; Wang, P.; Wang, S.; Wang, W.; Wu, S.; Xu, B.; Xu, J.; Yang, A.; Yang, H.; Yang, J.; Yang, S.; Yao, Y.; Yu, B.; Yuan, H.; Yuan, Z.; Zhang, J.; Zhang, X.; Zhang, Y.; Zhang, Z.; Zhou, C.; Zhou, J.; Zhou, X.; and Zhu, T. 2023. Qwen Technical Report. *arXiv preprint arXiv:2309.16609*.

Bain, M.; Nagrani, A.; Varol, G.; and Zisserman, A. 2021. Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1728–1738.

Borghi, A. M.; Falcinelli, I.; Fini, C.; Gervasi, A. M.; and Mazzuca, C. 2023. How Do We Learn and Why Do We Use Abstract Concepts and Words. *Frontiers in Young Minds*, 11: 1138574.

Borsos, Z.; Marinier, R.; Vincent, D.; Kharitonov, E.; Pietquin, O.; Sharifi, M.; Roblek, D.; Teboul, O.; Grangier, D.; Tagliasacchi, M.; and Zeghidour, N. 2023. AudioLM: a Language Modeling Approach to Audio Generation. arXiv:2209.03143.

Brooks, T.; Peebles, B.; Holmes, C.; DePue, W.; Guo, Y.; Jing, L.; Schnurr, D.; Taylor, J.; Luhman, T.; Luhman, E.; Ng, C.; Wang, R.; and Ramesh, A. 2024. Video generation models as world simulators.

Cai, Z.; Cao, M.; Chen, H.; and et al. 2024. InternLM2 Technical Report. arXiv:2403.17297.

Chen, H.; Zhang, Y.; Cun, X.; Xia, M.; Wang, X.; Weng, C.; and Shan, Y. 2024a. VideoCrafter2: Overcoming

Data Limitations for High-Quality Video Diffusion Models. arXiv:2401.09047.

Chen, T.-S.; Siarohin, A.; Menapace, W.; Deyneka, E.; wei Chao, H.; Jeon, B. E.; Fang, Y.; Lee, H.-Y.; Ren, J.; Yang, M.-H.; and Tulyakov, S. 2024b. Panda-70M: Captioning 70M Videos with Multiple Cross-Modality Teachers. arXiv:2402.19479.

et al., M. 2023. Hotshot-XL.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Gonen, H.; Iyer, S.; Blevins, T.; Smith, N. A.; and Zettlemoyer, L. 2022. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*.

Gu, J.; Wang, S.; Zhao, H.; Lu, T.; Zhang, X.; Wu, Z.; Xu, S.; Zhang, W.; Jiang, Y.-G.; and Xu, H. 2023. Reuse and Diffuse: Iterative Denoising for Text-to-Video Generation. arXiv:2309.03549.

He, Y.; Yang, T.; Zhang, Y.; Shan, Y.; and Chen, Q. 2023. Latent Video Diffusion Models for High-Fidelity Long Video Generation. arXiv:2211.13221.

Klein, T.; and Nabi, M. 2023. miCSE: Mutual Information Contrastive Learning for Low-shot Sentence Embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6159–6177. Toronto, Canada: Association for Computational Linguistics.

Levy, I.; Bogin, B.; and Berant, J. 2022. Diverse demonstrations improve in-context compositional generalization. *arXiv preprint arXiv:2212.06800*.

Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; and Chen, W. 2021. What Makes Good In-Context Examples for GPT-3? *arXiv preprint arXiv:2101.06804*.

Lu, Y.; Bartolo, M.; Moore, A.; Riedel, S.; and Stenetorp, P. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Ma, X.; Wang, Y.; Jia, G.; Chen, X.; Liu, Z.; Li, Y.-F.; Chen, C.; and Qiao, Y. 2024. Latte: Latent Diffusion Transformer for Video Generation. *arXiv preprint arXiv:2401.03048*.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; and et al. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Qing, Z.; Zhang, S.; Wang, J.; Wang, X.; Wei, Y.; Zhang, Y.; Gao, C.; and Sang, N. 2024. Hierarchical Spatio-temporal Decoupling for Text-to-Video Generation. In *CVPR*.

Qiu, H.; Xia, M.; Zhang, Y.; He, Y.; Wang, X.; Shan, Y.; and Liu, Z. 2024. FreeNoise: Tuning-Free Longer Video Diffusion via Noise Rescheduling. arXiv:2310.15169.

Sorensen, T.; Robinson, J.; Rytting, C. M.; Shaw, A. G.; Rogers, K. J.; Delorey, A. P.; Khalil, M.; Fulda, N.; and Wingate, D. 2022. An information-theoretic approach to prompt engineering without ground truth labels. *arXiv preprint arXiv:2203.11364*.

Unterthiner, T.; van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; and Gelly, S. 2018. Towards Accurate Generative Models of Video: A New Metric & Challenges. *arXiv preprint arXiv:1812.01717*.

Wang, J.; Yuan, H.; Chen, D.; Zhang, Y.; Wang, X.; and Zhang, S. 2023a. ModelScope Text-to-Video Technical Report. arXiv:2308.06571.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Fan, Y.; Dang, K.; Du, M.; Ren, X.; Men, R.; Liu, D.; Zhou, C.; Zhou, J.; and Lin, J. 2024. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*.

Wang, W.; Yang, H.; Tuo, Z.; He, H.; Zhu, J.; Fu, J.; and Liu, J. 2023b. VideoFactory: Swap Attention in Spatiotemporal Diffusions for Text-to-Video Generation. *arXiv preprint arXiv:2305.10874*.

Wang, Y.; Chen, X.; Ma, X.; Zhou, S.; Huang, Z.; Wang, Y.; Yang, C.; He, Y.; Yu, J.; Yang, P.; Guo, Y.; Wu, T.; Si, C.; Jiang, Y.; Chen, C.; Loy, C. C.; Dai, B.; Lin, D.; Qiao, Y.; and Liu, Z. 2023c. LAVIE: High-Quality Video Generation with Cascaded Latent Diffusion Models. arXiv:2309.15103.

Wang, Y.; He, Y.; Li, Y.; Li, K.; Yu, J.; Ma, X.; Chen, X.; Wang, Y.; Luo, P.; Liu, Z.; Wang, Y.; Wang, L.; and Qiao, Y. 2023d. InternVid: A Large-scale Video-Text Dataset for Multimodal Understanding and Generation. *arXiv preprint arXiv:2307.06942*.

Wu, C.; Huang, L.; Zhang, Q.; Li, B.; Ji, L.; Yang, F.; Sapiro, G.; and Duan, N. 2021. GODIVA: Generating Open-DomaIn Videos from nAtural Descriptions. arXiv:2104.14806.

Wu, H.; Zhang, E.; Liao, L.; Chen, C.; Hou, J.; Wang, A.; Sun, W.; Yan, Q.; and Lin, W. 2023. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 20144–20154.

Yang, Z.; Teng, J.; Zheng, W.; Ding, M.; Huang, S.; Xu, J.; Yang, Y.; Hong, W.; Zhang, X.; Feng, G.; et al. 2024. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. *arXiv preprint arXiv:2408.06072*.

Zhang, D. J.; Wu, J. Z.; Liu, J.-W.; Zhao, R.; Ran, L.; Gu, Y.; Gao, D.; and Shou, M. Z. 2023. Show-1: Marrying Pixel and Latent Diffusion Models for Text-to-Video Generation. arXiv:2309.15818.

Zhao, Z.; Wallace, E.; Feng, S.; Klein, D.; and Singh, S. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, 12697–12706. PMLR.

Zheng, Z.; Peng, X.; Yang, T.; Shen, C.; Li, S.; Liu, H.; Zhou, Y.; Li, T.; and You, Y. 2024. Open-Sora: Democratizing Efficient Video Production for All.

A Annotation Algorithm

Algorithm 1: Annotation Algorithm for *AbsText2Video*

Input: ConcreteTexts C, Threshold τ, InitPrompt P_{init}, FixedPrompt P_{fix}, Interval $[lo, hi]$, Scaling α
Tools: LLMs $\{\mathcal{M}_k\}_{k=0}^{n-1}$, Embedder f, EvalSet $\langle C_{eval}, A_{eval} \rangle$
Output: AbstractTexts A

function GENECONCS(a, $\mathcal{M}$, P)
  return $\{\mathcal{M}(a, P)\}$ ▷ Parallelizable generation
end function
for $c \in C$ **do**
  Scores $\leftarrow \{\}$
  CCS $\leftarrow \{\}$
  for $k \leftarrow 0$ to $n-1$ **do**
    $a_k \leftarrow \mathcal{M}_k(c, P_{init})$
    $\sigma_k \leftarrow \cos(f(a_k), f(c))$
    Scores$[k] \leftarrow \sigma_k$
  end for
  if $\forall \sigma_k \notin [lo, hi]$ **then**
    $a^* \leftarrow a_{\arg\min \text{Scores}}$
  else
    for $k \leftarrow 0$ to $n-1$ **do**
      if $\sigma_k \in [lo, hi]$ **then**
        $\{c_j^{cyc}\} \leftarrow$ GeneConcs$(a_k, \mathcal{M}_k, P_{fix})$
        Var$_k \leftarrow \mathbb{V}\text{ar}(\{f(c_j^{cyc})\})$
        $\sigma_k^{cyc} \leftarrow \max_j \cos(f(c_j^{cyc}), f(c))$
        CCS$_k \leftarrow \sigma_k + \alpha \cdot$ Var$_k + \sigma_k^{cyc}$
        CCS$[k] \leftarrow$ CCS$_k$
      end if
    end for
    $a^* \leftarrow a_{\arg\max \text{CCS}}$
    $A \leftarrow A \cup \{a^*\}$
    if $\max(\text{CCS}) \geq \tau$ **then**
      $P_{new} \leftarrow$ Prune$(P_{init} \cup \{(c, a^*)\})$
      if EVAL$(P_{new}) >$ EVAL(P_{init}) **then**
        $P_{init} \leftarrow P_{new}$
      end if
    end if
  end if
end for
return A

B More Examples

Figure 5: More examples from the *AbsText2Video* dataset.

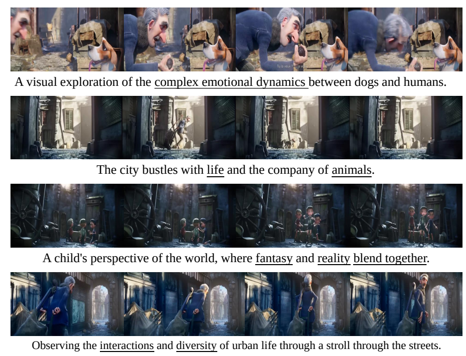


Figure 6: More examples from the *AbsText2Video* dataset.

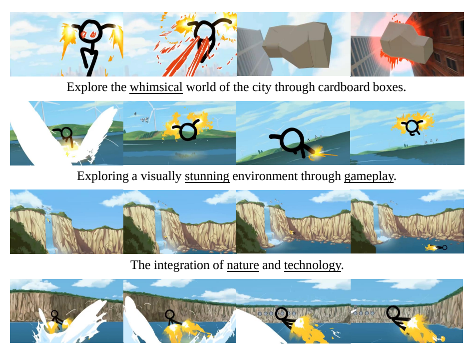


Figure 7: More examples from the *AbsText2Video* dataset.