Robustness Analysis of Video-Language Models Against Visual and Language Perturbations (Supplementary)

https://bit.ly/3CNOly4

Madeline C. Schiappa
University of Central Florida
madelineschiappa@knights.ucf.edu

Shruti Vyas
University of Central Florida
shruti@crcv.ucf.edu

Hamid Palangi
Microsoft Research
hpalangi@microsoft.com

Yogesh S. Rawat*
University of Central Florida
yogesh@crcv.ucf.edu

Vibhav Vineet*
Microsoft Research
vivineet@microsoft.com

Contents

*The authors contributed equally as supervisors to this paper.

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

1 Implementation Details

1.1 Visual Perturbations

Below are more details on the perturbations applied to videos. Examples of these perturbations can also be found at `https://bit.ly/3CNOly4`.

Noise These perturbations apply transformations at the pixel level of each frame in a video. The different noises are *Impulse*, *Gaussian*, *Shot*, and *Speckle*. **Impulse** noise simulates corruptions cause by bit errors by applying a combination of salt and pepper noise with amounts ranging from .03, .06, .09, 0.17, 0.27. **Gaussian** noise simulates low-lighting conditions by first normalizing the pixel values then adds a random normal noise scaled at values .08, .12, 0.18, 0.26, 0.38 based on severity. *Shot* noise simulates electronic noise caused by the discrete nature of light by applying a combination of salt and pepper noise with amounts ranging from .03, .06, .09, 0.17, 0.27. *Speckle* noise simulates additive noise and is similar to Gaussian but where the random value is then multiplied by the normalized pixel value.

Blur Blur perturbations apply transformations that simulate camera motion and focus. *Motion* blur increases the radius and sigma of the kernel which is used to create the motion blurring effect ranging from (10, 3), (15, 5), (15, 8), (15, 12), and (20, 15) based on severity. *Zoom* blur blurs towards the center of the frame while increasing the zoom factor based on severity. *Defocus* blur imitates a defocused lens over the entire frame. We increase the radius of the disk which is convolved over the image to create defocus blurring effect ranging from (3, 0.1), (4, 0.5), (6, 0.5), (8, 0.5), (10, 0.5) based on severity.

Digital *JPEG* compression converts each frame to a JPEG with quality ranging from 25, 18, 15, 10, 7 based on severity. *MPEG1* compresses the original video to using the ffmpeg [18] format mpeg2video with levels ranging from 20, 40, 60, 80, 100. *MPEG2* compresses the original video to using the ffmpeg format mpeg4 with levels ranging from 15, 30, 45, 60, 75. This compression tends to actually affect the playing of the video, where frames are missing and/or skipped. These slight frame changes allows these perturbations to be considered temporal as well. This can be seen in an example in Figure 2 under Digital where the frame does not perfectly align with the frames for the other perturbations because it is slightly off temporally. We can consider these perturbations as spatio-temporal as they alter both spacial features and temporal features.

Temporal *Jumbling* splits a video into segments of lengths 32, 16, 8, 4, and 2 where the higher is less severe and lower is more severe. The frames within each segment are then randomly shuffled. *Box Jumbling* splits a video into segments of lengths 4, 9, 16, 25, 36 where the higher is more severe and lower is less severe. The segments are then randomly shuffled. *Sampling* transforms a video from the original method's frames per second (FPS) to keep consistency with the original approach. However, it slows the playback speed by sampling frames uniformly at a varying level of rates 2, 4, 8, 16, and 32, where the higher is more severe. *Reverse Sampling* is the same as sampling but also reverses the video after sampling. *Freeze* This perturbation will choose a percentage of frames to select, ranging from 40%, 20%, 10%, 5% and 2.5%. The more frames selected, the less severe the perturbation. These selected frames are then repeated until they reach the next sequential frame, simulating a frozen live stream video.

Camera These perturbations simulate irregularities with camera motion and include *Static rotation*, *Rotation* and *Translation*. Static rotate rotates every frame the same degree, Rotation rotates each frame by a random degree, and Translation randomly chooses a new center in the frame to crop to for each frame as if the camera is randomly shaking.

1.2 Text Perturbations

In Table 1 there are examples for each perturbation when the input text is "a little girl does gynmanstics" from the MSRVTT dataset. This section discusses the implementation of each in more detail.

ChangeChar Perturbations, natural and machine-based, that alter a character in one or several of the words in the text. For natural-based perturbations, this includes *SpellingError*, *Keyboard*,

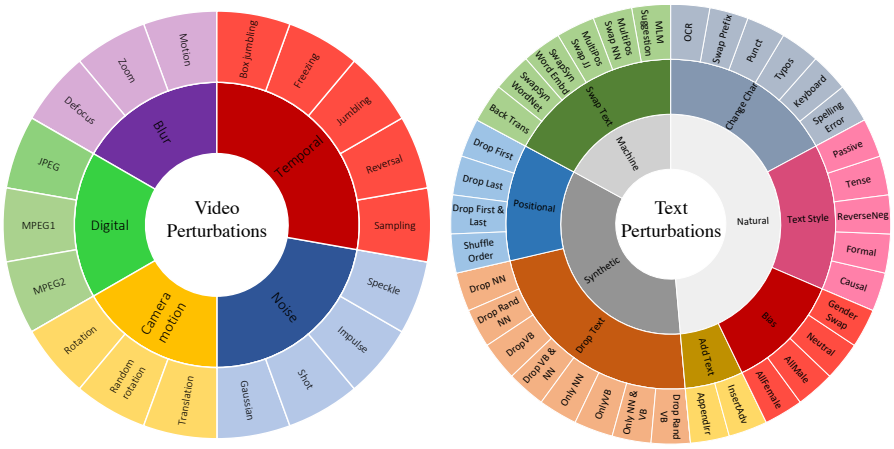

Figure 1: The summary of how this paper organizes both visual and text perturbations used to evaluate on the text-to-retrieval task for multimodal models. On the visual side, each perturbation also ranges from a severity of 1 to 5.

and *Typos* [19]. These are based on common spelling errors, keyboard mistypes, and general typos. Typos for example randomly inserts, deletes, swaps or replaces a single letter within one word while keyboard alters text by common keyboard mistakes such as "word → to work". Machine-based perturbations include *OCR*, *SwapPrefix* and *Punctuation* [19]. SwapPrefix for example will swap the prefix of one word while keeping its part-of-speech tag. Punctuation appends and/or prepends random punctuation to the sentence and OCR uses random values to stimulate an OCR, or optical character recognition, error.

SwapText It is machine-based perturbations that swap word(s) from the original text. Several perturbations that make word swaps based on text models include *BackTrans* which replaces text with phrases by using back-translation [19]. *SwapSynWordNet* and *SwapSynWordEmbedding* both swap a words with their synonyms as determined by either WordNet [7] or by Glove [14]. *MLM suggestion* swaps one syntactic category element of the sentence with what would be predicted by masked language models (MLM) [19]. *MultiPosSwapJJ* and *MultiPosSwapNN* replaces adjectives and nouns respectively with words holding multiple parts-of-speech (POS).

AddText Perturbations [19] that are natural-based and add text to the original. *AppendIrr* appends irrelevant phrases to the original text while *InsertAdv* adds an adverb before each verb.

TextStyle Perturbations that are natural-based and change the style of the text. These include *Tense*, *Passive*, *Casual*, *Formal*, and *ReverseNeg* [5]. The perturbations Passive, Casual, and Formal change the text style to those specific styles. Tense changes the tense of the text and ReverseNeg reverse negates the original text.

Bias Perturbations that are natural-based that change the gender of a given phrase. These vary in *AllFemale*, *AllMale*, *GenderSwap*, and *Neutral*. The netural perturbation removes female and male references and replaces them with neutral ones. For example, a reference to "a man" and "a woman" would be replaced with "a person". GenderSwap swaps male references with female and vice versa using [16].

DropText Perturbations are synthetic and drop words based on their part-of-speech (POS) tag. *DropNN*, *DropVB*, and *DropVBNN* are different variations of dropping words based on whether the POS tags are Noun and/or Verb. *OnlyNN*, *OnlyVB*, and *OnlyNNVB* drops all words but those with POS NN and/or VB. *RandNN* and *RandVB* drop one random noun/verb. This is done using the NLTK [3] package to first extract POS tags for each word. Using these POS tags, based on the respective perturbation, words are "dropped" by replacing them with "[UNK]" in order to maintain the original phrase length.

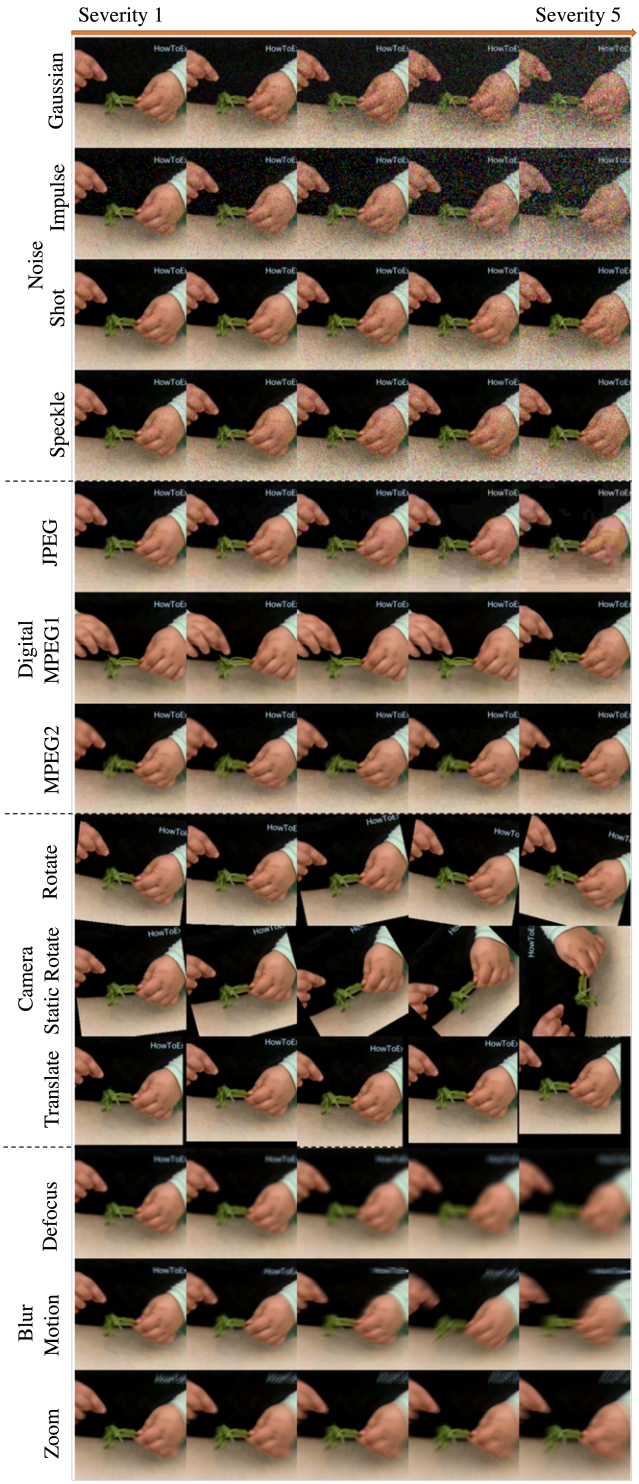

Figure 2: Visualizations of each perturbation for a single frame in a video from the YouCook2 dataset. Severity increases from left to right for each perturbation.

Positional These include *DropFirst*, *DropLast*, *DropFirstandLast*, and *ShuffleOrder*. Drop-related perturbations will replace a word at that position with an "[UNK]" tag. The ShuffleOrder perturbation

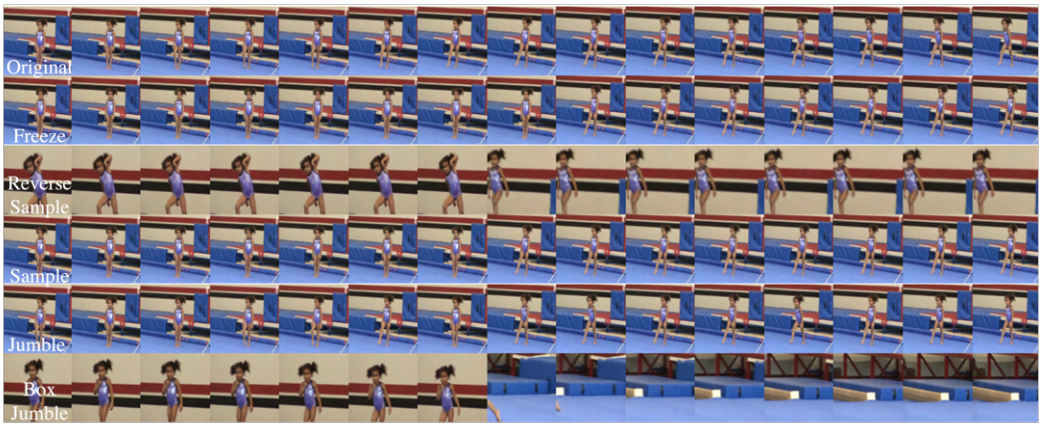

Figure 3: A visualization of the temporal perturbations for a video showing "a little girl does gymnastics".

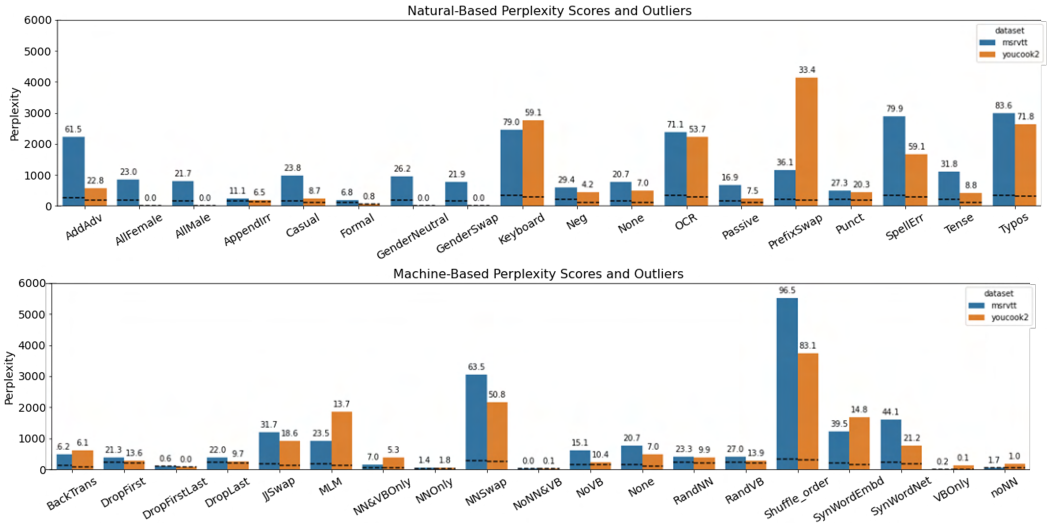

Figure 4: The perplexity scores for the different text perturbations using GPT-3. The bars represent the average perplexity for the entire corpus, the dashed lines represent the perplexity when removing outliers based on a threshold of a 500, and the numbers atop the bars are the percent of outliers that are removed when using the threshold.

shuffles the words in a phrase randomly. More details on the generated text perturbation are provided in the Appendix.

1.3 Analysis of Perturbed Text

To understand the severity of each perturbation we evaluate the perturbed text that is generated using multiple metrics including *perplexity*, *BLEU*, *METEOR* and *ROUGE*.

Perplexity of Perturbations We first look at the perplexity measurement which measures the probability of a sentence using the large text model GPT-3 [4]. For each word in the sentence, the probability of the next word being present is measured and if the probability of the next word is low, then the perplexity for that sentence will be higher. Perplexity is not necessarily a good measurement for quality, but it is useful for measuring how statistical models may struggle with text. We first observe further support on how different the two datasets are on the text side where the perplexity of MSRVTT is 762.97 and for YouCook2 480.84.

Table 1: Examples of all text perturbations in each category fot the text "a little girl does gymnastics" from the MSRVTT dataset.

Type	Perturbation	Perturbed
AddText	AddAdv	a little girl specifically does gymnastics
	AppendIrr	On this occasion, a little girl does gymnastics
Bias	AllFemale	a little girl does gymnastics
	AllMale	a little boy does gymnastics
	GenderNeutral	a little child does gymnastics
	GenderSwap	a little boy does gymnastics
ChangeChar	Keyboard	a little girl dofs gymnastics
	OCR	a little girl does gymnastic8
	PrefixSwap	a little girl does gymnastics
	Punct	" a little girl does gymnastics, "
	SpellErr	a lettil girl does gymnastics
	Typos	a little girl des gymnastics
DropText	NN&VBOnly	[UNK] [UNK] girl does gymnastics
	NNOnly	[UNK] [UNK] girl [UNK] gymnastics
	NoNN&VB	a little [UNK] [UNK] [UNK]
	NoVB	a little girl [UNK] gymnastics
	RandNN	a little girl does [UNK]
	RandVB	a little girl [UNK] gymnastics
	VBOnly	[UNK] [UNK] [UNK] does [UNK]
	NoNN	a little [UNK] does [UNK]
Positional	DropFirst	[UNK] little girl does gymnastics
	DropFirstLast	[UNK] little girl does [UNK]
	DropLast	a little girl does [UNK]
	ShuffleOrder	a girl gymnastics does little
SwapText	BackTrans	a little girl gymnastics
	JJSwap	a anodyne girl does gymnastics
	MLM	a teenage girl does gymnastics
	NNSwap	a little output does gymnastics
	SynWordEmbd	a good girl does gymnastics
	SynWordNet	a little girl manage gymnastics
TextStyle	Casual	A little girl that does gymnastics
	Formal	A young woman does gymnastics.
	Neg	a little girl does not gymnastics
	Passive	gymnastics is done by a little girl
	Tense	a little girl did gymnastics

The results of this analysis is shown in Figure 4 where machine-based perturbations are the bottom row and natural-based perturbations are the top. Between natural and machine-based there are no obvious differences in perplexity overall, both appear to have challenging distribution shifts according to the perplexity of GPT-3. Changing characters in words appear to result in higher perplexity consistently across the different variations of *ChangeChar* perturbations. PrefixSwap and NNSwap are additionally high in perplexity on the machine-based side. These results would indicate the statistical models should struggle most with character changes and word swaps or drops based on POS-tags. The most perplex version of text is when words are shuffled in *ShuffleOrder*, as the words positions to each other are no longer meaningful. In summary, machine-learning based approaches *are likely to struggle most with character swapping perturbations and shuffling of words*.

Comparison Metrics to Original Text We also compare the perturbed text to the original text using the traditional metrics, BLEU [13], METEOR and Rouge. The results for these are averaged across the different perturbations for each type for both the MSRVTT and YouCook2 datasets in Table 2. Perturbations that *DropText* are most dissimilar to the original text for both datasets. Depending on the dataset, *AddText*, *TextStyle* and *ChangeChar* are similarly dissimilar to the original text, meaning

Table 2: Distribution Shift evaluation on MSRVTT and YouCook2 captions respectively.

MSRVTT	AddText	Bias	ChangeChar	DropText	Positional	SwapText	TextStyle
BLEU4	0.57	0.88	0.60	0.29	0.64	0.68	0.56
Meteor	0.76	0.94	0.79	0.53	0.78	0.87	0.80
ROUGE-l F1	0.16	0.22	0.22	0.17	0.21	0.21	0.21
YouCook2	AddText	Bias	ChangeChar	DropText	Positional	SwapText	TextStyle
BLEU4	0.64	—	0.58	0.34	0.62	0.66	0.65
Meteor	0.76	—	0.78	0.52	0.76	0.85	0.81
ROUGE-l F1	0.17	—	0.23	0.17	0.22	0.22	0.22

models should be robust but show some level of performance reduction. The most similar is *Bias*, meaning models should be highly robust to Bias.

In summary, these scores indicate that we have a varying level of difficulty with our text perturbations across categories, allowing for variable securities of distribution shift. The *most challenging is DropText* and *the least challenging is Bias*.

1.4 Model Implementations

To process data, train and evaluate models, we used our internal cluster with single-GPU use per run. All models except the MIL NCE [11] and FIT [1] use features extracted from the visual encoder of the MIL NCE. For MIL NCE, the clips were split into 4 clips of 32 frames each. For VideoClip [20] and COOT [8] input videos of size 224×224 at 30 fps were fed to the pre-trained S3D model where we extracted features at the final layer with an embedding size of 512. The same procedure was used for UniVL [10] with the difference being the features are extracted at the earlier layer "mixed5c" with an embedding size of 1024. For perturbations, we first perturb the video before extracting S3D features, therefore we have collect perturbed S3D features for embedding sizes 512 and 1024 for each perturbation and severity. We used the original code for these models to extract features to ensure that the procedure is the same as the original authors'. These original feature extraction scripts are located in the Github repositories [9] and [15]. FIT [1] splits each clip into 4 segments and randomly selects a single frame from each segment.

2 Limitations

This study has several limitations which include 1) the use of simulated noise due to the challenge of obtaining real-world data and 2) the analysis is on a limited number of models due to availability of usable code and model weights. The analysis provided is limited to the models we have used in this study. We tried our best to benchmark all publicly available models that provided weights and used both text and video. Further challenges from each approach included 1) having strict requirements on large data pre-processing, 2) requiring heavy GPU usage for both testing and even more so for training, and 3) having their own specific testing bed. These factors limited both the selection and time it took to implement models. In future, hopefully a larger number of models will be available.

3 Licensing

All the models which we have used in this study are available in public domain. The model code for HowTo100m MIL [12] and COOT [8] have the Apache 2.0 license and the model code for VideoClip [20] and UniVL [10] has the MIT license and are publicly available. We will provide YouCook2-P and MSRVTT-P publicly for future research. These datasets are based on existing YouCook2 [21] and MSRVTT [17] datasets and we are not using any new video sources. Both these datasets are available in public domain for research purposes and therefore similar licensing is applicable to the newly curated datasets.

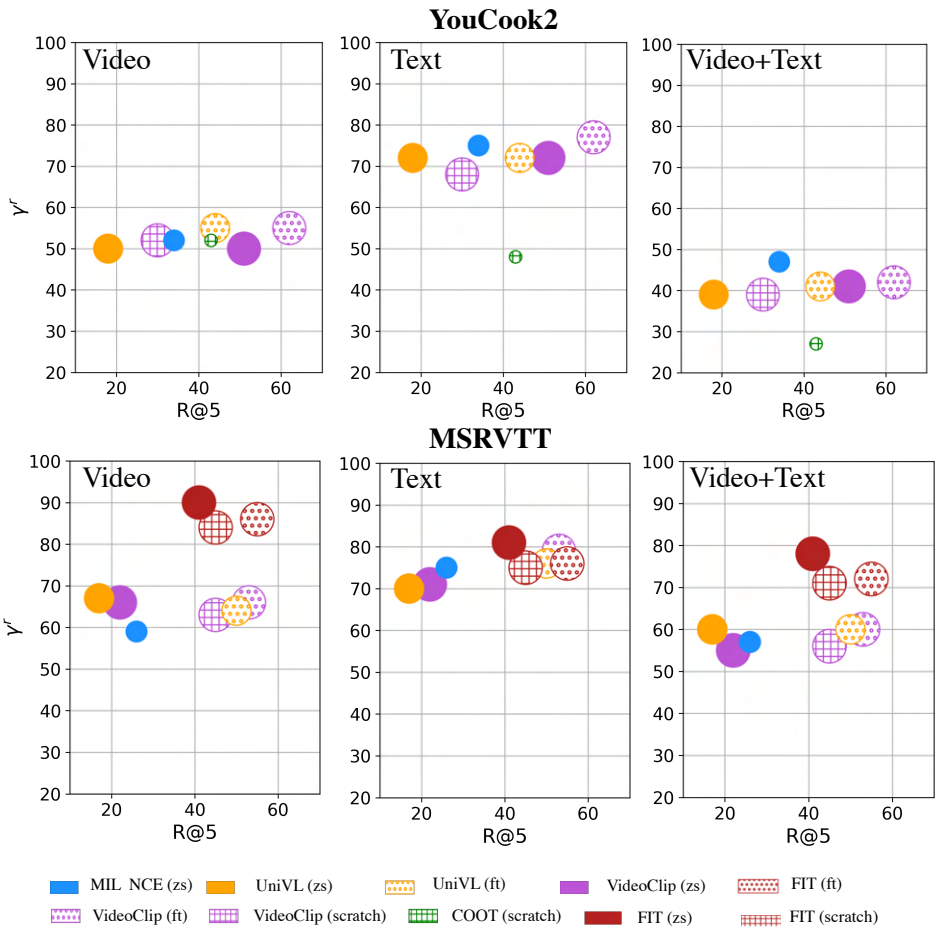

Figure 5: On the x-axis, R@5 for text-to-video retrieval and y-axis is the drop in performance when data is perturbed measured by relative robustness γ^r for both datasets. These results are aggregated across all categories for the modality and all combinations we generated for the combination of video and text.

4 Impact

To our understanding, there are no negative societal impacts of our work. The goal of this work was to evaluate the robustness of models that may later be used in real-world settings. We aimed to improve the societal impacts by evaluating these models on real-world distribution shifts including potential bias in text.

5 Additional Results

Figure 5 shows the relative robustness for the text-to-video retrieval task at R@5 aggregated across all categories for video, text and when both are perturbed against the performance. The results vary based on the dataset due to MSRVTT focusing on short videos of activity and YouCook2 breaking up a long-complex activity into shorter clips. The differences in model robustness and performance for the different datasets *indicates a difference on how models handle clips from long, complex activities* compared to videos that are short and of a simple activity. On the longer activity dataset YouCook2, pre-training is noticeably more important for both robustness and performance.

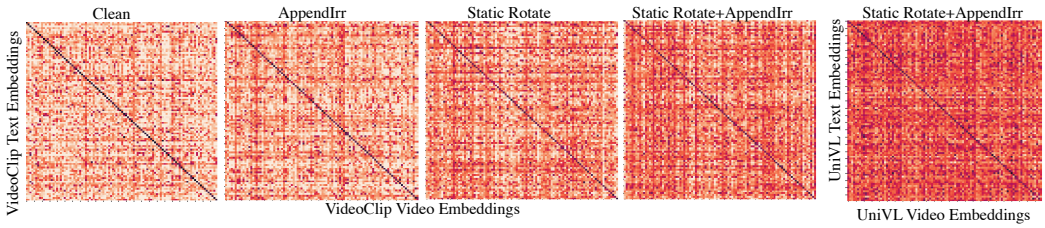

Figure 6: Similarity matrices where the x-axis are video representations and the y-axis are text representations sampled on the YouCook2 dataset. The darker the color, the more similar. When both video and text are perturbed, a compounding effect is shown by the increase in similarity for samples that do not match. Additionally, VideoClip shows less similarity between incorrect pairs when both are perturbed as compared to UniVL which utilizes cross-attention.

5.1 Analysing Feature Space

To visualize the changes to the embedding space when text and video are perturbed, we selected videos that were accurately matched to their respective text in the baseline and observed their similarity change when perturbations were made. Figure 6 visualizes the baseline, when text is perturbed with appending irrelevant phrases, when video is perturbed by a consistent rotation and when both these perturbations are applied. As these perturbations are added, the similarity between video and other text begins to increase. When both video and text are perturbed, this effect is most visible. Additionally, the UniVL model which uses cross-attention shows even more similarity between video and text when both are perturbed. This does not necessarily mean that UniVL is less robust in this case, but it can indicate with cross-attention, video and text are generally more similar and the difference between ranking one video to the other is much smaller than in other models.

Figures 7 and 8 show tsne plots of the feature space for the Videoclip and UniVL models respectively when pre-trained and not fine-tuned. The colors indicate the recipe type and are just a way of visualizing space that should be more similar than videos and text from other recipes. It is important to note that the models are not trained on creating a space that clusters recipes together; therefore using recipes is just an arbitrary way of visualizing this space. In Figure 7, we see that when one modality is perturbed, the embedding space of the other is relatively untouched. When both video and text are perturbed, both spaces are impacted. In Figure 8, *we see a similar trend, where when one modality is perturbed, the others embedding space is relatively untouched, even though their is use of cross-attention between the two modalities.*

5.2 Breakdown of Perturbations

Results for each perturbation when text is perturbed are shown in Figure 9 and 10. In these figures, the black, dashed line indicate the original performance and the bars represent the performance when text is perturbed. The larger the difference between the top of the bar and the horizontal line indicate a larger drop in relative performance. *The figures visualize the noticeable drop in performance for ChangeChar, a perturbation humans are highly robust to. It also shows how robust models are to AddText, Positional and Text Style. Finally, it shows models are surprisingly robust to the synthetic noise of Drop Text.* Figure 11 and 12 show performance R@5 over the varying levels of severity. On both datasets, the majority of models are not robust to spacial perturbations such as Noise and Blur. *FIT [1], which uses a ViT [6, 2] visual encoder, is noticeably robust to spatial noise.* When text is aligned with segments, models are robust to temporal perturbations but are not on YouCook2 when text is no longer aligned with its segment from the long, complex video. However, *frame order does not matter within the texts respective segment*. Models are *surprisingly robust against spatio-temporal, or Digital, perturbations, struggling most with JPEG compression*.

5.3 Absolute Robustness

The absolute robustness scores for text perturbations are shown in Table 3 and visual perturbations in Table 4. When observing absolute robustness, UniVL Align [10], pre-trained but not fine-tuned, is the typically the most robust model that uses a CNN visual encoder, while FIT [1] is the most robust

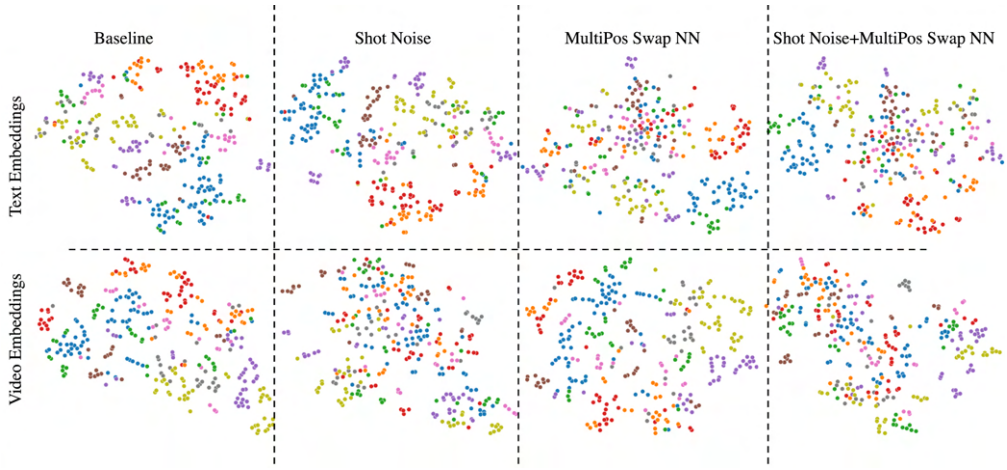

Figure 7: TSNE visualizations for output of the VideoClip model for text and video with different perturbations where colors are recipe types. This is a visualization that uses TSNE to compress the high-dimensional feature space to 2D space. Using this, we observe that when one modality is perturbed, the embedding space of the other is untouched. When both video and text are perturbed, both embedding spaces are impacted.

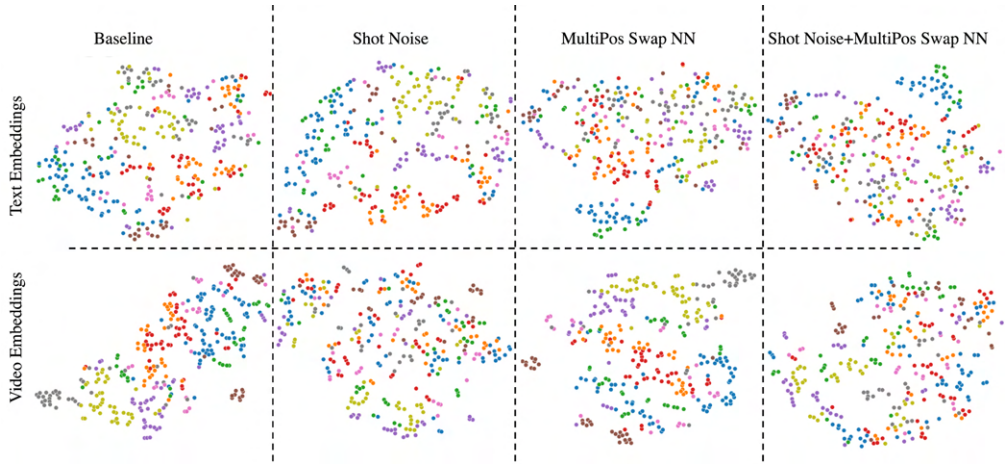

Figure 8: TSNE visualizations for output of the UniVL model for text and video with different perturbations where colors are recipe types. This is a visualization that uses TSNE to compress the high-dimensional feature space to 2D space. When one modality is perturbed, the embedding space of the other is untouched, despite cross-attention being used. When both video and text are perturbed, both embedding spaces are impacted.

when video is perturbed of all models. UniVL uses a cross-encoder architecture with an alignment based objective function. This differs from the relatively robust results in which it varies which model and pre-training strategy is more robust. In Table 4, the models struggle most with Blur, Noise and Digital. *Digital is likely challenging because it perturbs both spacial features and temporal* (see Figure 2).

5.4 Perturb Modality

Table 5 shows the relative and absolute robustness across the video, text or both being perturbed for each dataset. FIT [1] is noticeably more relatively robust on video perturbations, likely due to the use of ViT as the visual encoder. COOT [8] is noticeably low at text robustness, likely due to using pre-extracted text features instead of training a text encoder. When both is perturbed, UniVL has the highest absolute robustness for both datasets. UniVL uses cross-attention during training as opposed

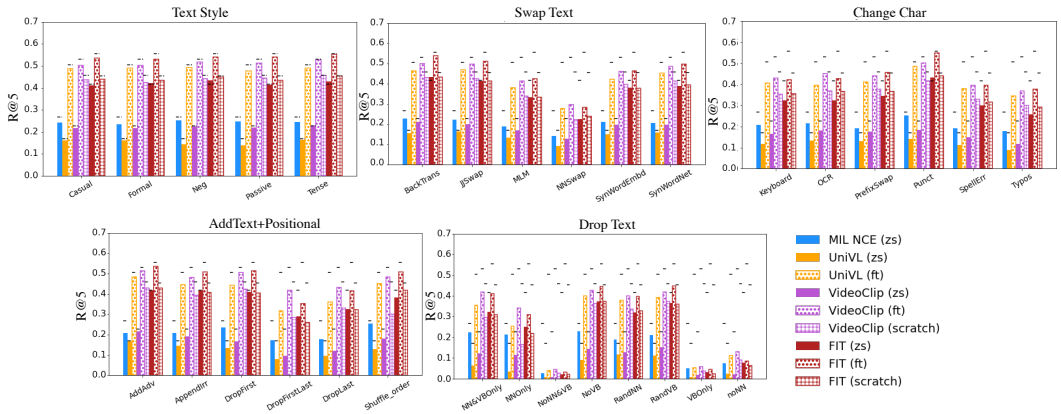

Figure 9: Perturbation specific results for text perturbations on the MSRVTT dataset. The black, horizontal lines indicate the retrieval on the clean dataset R_c while the bar indicates the retrieval on the perturbed dataset R_p. Fine-tuned models are the highest performers but are not always more robust.

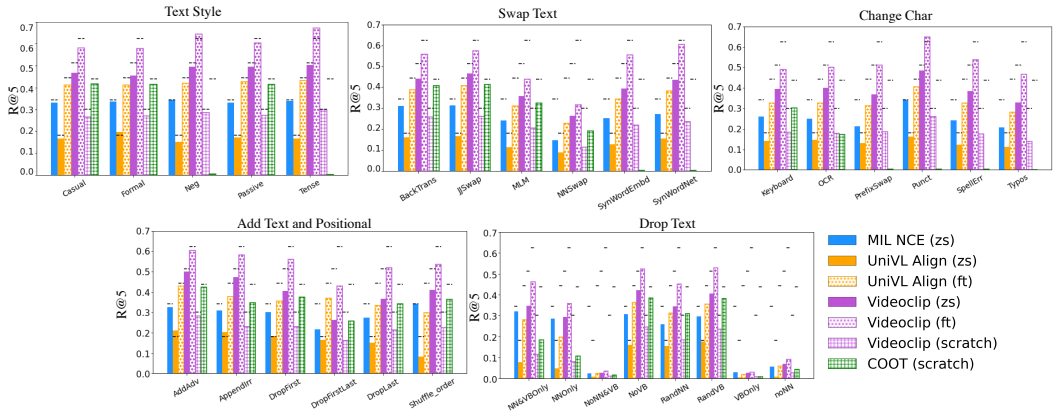

Figure 10: Perturbation specific results for text perturbations on the YouCook2 dataset. The black, horizontal lines indicate the retrieval on the clean dataset R_c while the bar indicates the retrieval on the perturbed dataset R_p.

to a two-encoder architecture like VideoClip. This could be a potential reason for the high absolute robustness UniVL demonstrates when using zero-shot evaluation.

5.5 MRSVTT QA

To understand if these findings transcend to other video-language tasks, we evaluated VideoClip on the multiple choice VideoQA task with results in Table 6 and 7.When only text is perturbed, the the difference between pre-training strategy is not consistent unlike the text-to-video retrieval task. Unlike the previous task, zero-shot typically is the least robust and scratch is as robust or more than fine-tuning. This indicates that when the task is between smaller candidates, pre-training on a large corpus of data may be less necessary for both performance and robustness.

When text is perturbed, the zero-shot model is typically more robust. Very different findings between the two modalities. This is likely because this task is similar to video-to-text retrieval and we have already observed models are less robust to visual perturbations. Zero-shot therefore is more relatively robust because of the nature of the pre-training dataset of HowTo100m [12] being a variety of noisy YouTube videos.

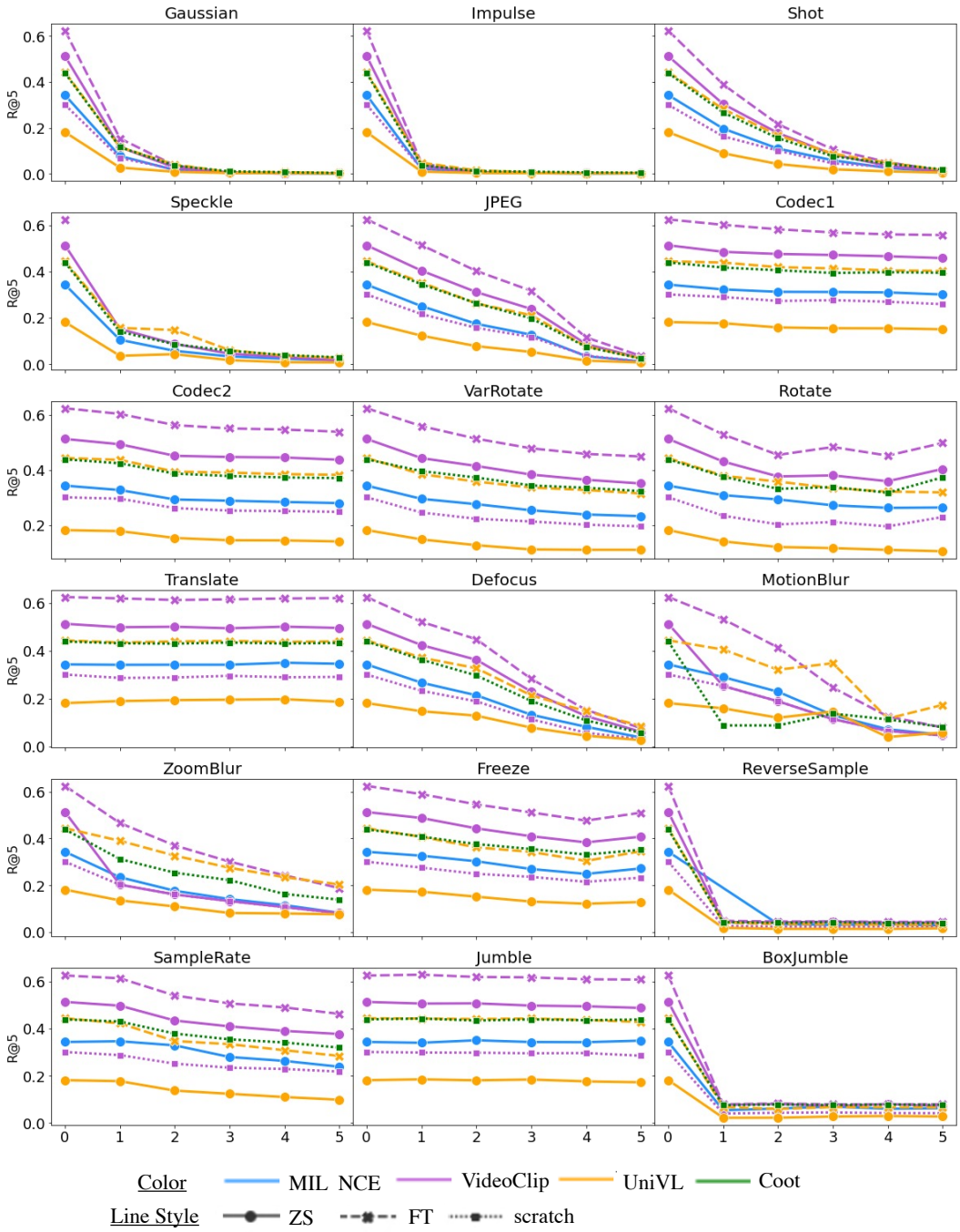

Figure 11: Performance R@5 when video is perturbed for different levels of severity on the YouCook2 dataset. Models are less robust against spacial perturbations and strongly perturbed against Temporal perturbations that maintain alignment between text and segments. When alignment is disturbed, models are no longer robust.

5.6 Natural vs. Machine vs. Artificial Text Perturbations

To understand models based on the sub-categories of natural, machine-learning based and synthetic, we aggregate scores these categories in Table 8 (see Figure 1 for more details on these categories). Synthetic perturbations are those in DropText and Positional as they would be rare occurrences in the real world, although still possible with automatic speech recognition for example. When only text is

Table 3: Average Absolute robustness scores and their standard deviations $\gamma^a \pm \sigma$ for each category of distribution shifts for text perturbations. The UniVL model is typically the highest performer. Overall, models are very robust to text perturbations when considering the absolute score.

MSRVTT$\gamma^a \pm \sigma$	AddText	Bias	ChangeChar	DropText	Positional	SwapText	TextStyle
FIT (scratch)	0.96±0.02	0.93±0.03	0.90±0.05	0.76±0.15	0.90±0.07	0.91±0.07	0.99±0.01
VideoClip (scratch)	0.95±0.03	0.94±0.02	0.90±0.04	0.75±0.15	0.88±0.06	0.91±0.08	0.98±0.01
FIT (zs)	**1.00±0.00**	0.98±0.02	0.91±0.06	0.80±0.15	**0.94±0.05**	0.95±0.08	**1.01±0.01**
MIL NCE (zs)	0.94±0.00	0.97±0.01	0.94±0.03	0.88±0.09	**0.94±0.04**	0.93±0.03	0.98±0.01
UniVL (zs)	0.99±0.02	**1.00±0.01**	**0.95±0.02**	**0.89±0.05**	**0.94±0.03**	**0.97±0.03**	0.98±0.01
VideoClip (zs)	0.98±0.02	0.99±0.01	0.94±0.03	0.86±0.06	0.91±0.04	0.96±0.03	0.99±0.01
FIT (ft)	0.97±0.02	0.94±0.03	0.88±0.06	0.72±0.19	0.89±0.08	0.90±0.09	0.98±0.01
UniVL (ft)	0.96±0.03	0.94±0.03	0.90±0.05	0.74±0.16	0.89±0.06	0.91±0.07	0.98±0.01
VideoClip (ft)	0.97±0.02	0.95±0.02	0.90±0.05	0.75±0.17	0.93±0.04	0.91±0.08	0.98±0.01

YouCook2$\gamma^a \pm \sigma$	AddText	Bias	ChangeChar	DropText	Positional	SwapText	TextStyle
COOT (scratch)	0.95±0.05	—	0.64±0.13	0.74±0.16	0.90±0.05	0.78±0.19	0.81±0.23
VideoClip (scratch)	0.96±0.04	—	0.89±0.04	0.81±0.10	0.91±0.03	0.91±0.05	0.98±0.01
MIL NCE (zs)	0.97±0.01	—	0.91±0.05	0.85±0.14	0.94±0.05	0.91±0.06	**0.99±0.00**
UniVL (zs)	**1.03±0.01**	—	**0.95±0.02**	**0.90±0.07**	**0.96±0.04**	**0.95±0.03**	**0.99±0.02**
VideoClip (zs)	0.97±0.02	—	0.88±0.05	0.73±0.17	0.85±0.07	0.88±0.07	0.97±0.02
UniVL (ft)	0.96±0.04	—	0.89±0.04	0.76±0.15	0.90±0.03	0.90±0.07	0.98±0.01
VideoClip (ft)	0.97±0.02	—	0.90±0.07	0.69±0.22	0.89±0.06	0.88±0.11	**0.99±0.04**

Table 4: Average Absolute robustness scores and their standard deviations $\gamma^a \pm \sigma$ for each category of distribution shifts for video perturbations. The UniVL model is typically the most robust model. Models are least robust to Noise, Blur and Digital.

MSRVTTγ^a	Blur	Camera	Digital	Noise	Temporal
FIT (scratch)	0.85±0.06	0.96±0.05	0.93±0.03	0.88±0.11	1.00±0.01
VideoClip (scratch)	0.79±0.11	0.91±0.08	0.78±0.10	0.65±0.09	0.98±0.02
FIT (zs)	0.91±0.05	**0.99±0.04**	**0.95±0.03**	**0.92±0.09**	**1.01±0.01**
MIL NCE (zs)	0.89±0.05	0.94±0.03	0.84±0.05	0.80±0.05	0.97±0.02
UniVL (zs)	**0.93±0.04**	0.98±0.02	0.93±0.03	0.88±0.03	0.99±0.01
VideoClip (zs)	0.91±0.05	0.96±0.03	0.91±0.04	0.82±0.04	0.99±0.01
FIT (ft)	0.86±0.06	0.96±0.06	0.91±0.04	0.87±0.11	1.00±0.01
UniVL (ft)	0.80±0.10	0.92±0.06	0.79±0.09	0.63±0.11	0.95±0.04
VideoClip (ft)	0.78±0.12	0.91±0.07	0.80±0.10	0.61±0.11	0.97±0.02

YouCook2γ^a	Blur	Camera	Digital	Noise	Temporal
COOT (scratch)	0.74±0.09	0.94±0.04	0.88±0.13	0.62±0.07	0.82±0.17
VideoClip (scratch)	0.83±0.07	0.94±0.04	0.91±0.09	0.73±0.05	0.87±0.12
MIL NCE (zs)	0.81±0.08	0.95±0.04	0.90±0.10	0.70±0.05	0.87±0.13
UniVL (zs)	**0.91±0.04**	**0.96±0.04**	**0.94±0.06**	**0.84±0.02**	**0.92±0.07**
VideoClip (zs)	0.66±0.11	0.91±0.06	0.87±0.15	0.54±0.08	0.78±0.20
UniVL (ft)	0.82±0.10	0.93±0.05	0.89±0.13	0.62±0.07	0.80±0.17
VideoClip (ft)	0.67±0.16	0.91±0.07	0.85±0.18	0.45±0.11	0.73±0.25

perturbed, UniVL on zero-shot learning is typically more relatively robust for all three categories, although close to the MIL NCE robustness on synthetic perturbations.

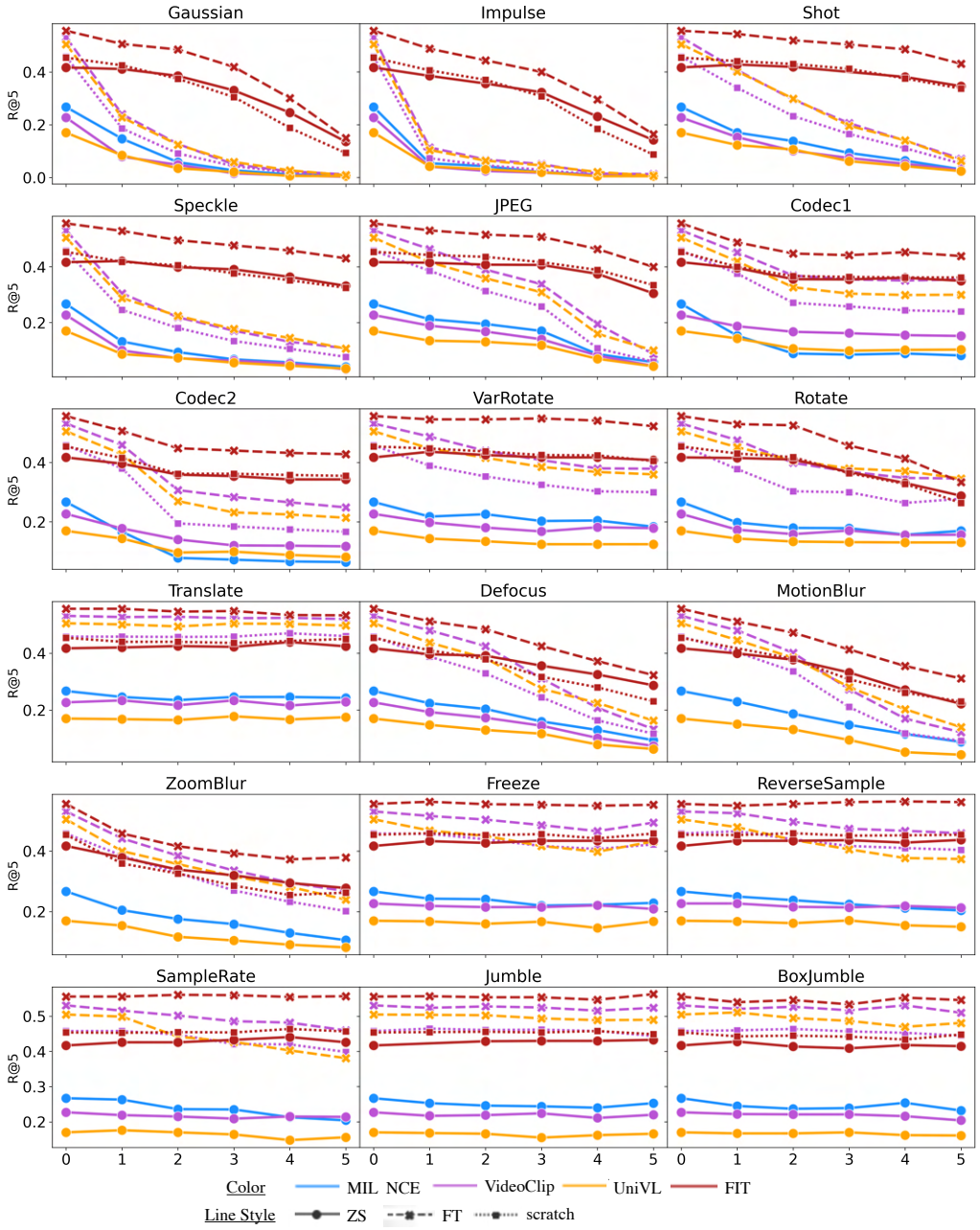

Figure 12: Performance R@5 when video is perturbed for different levels of severity on the MSRVTT dataset. Models are less robust against spacial perturbations and strongly perturbed against Temporal perturbations. Models are surprisingly robust against spatio-temporal (Digital) perturbations, struggling most with JPEG.

Table 5: Average Absolute robustness γ^a, Relative Robustness scores γ^r and their standard deviations $\pm\sigma$ across video, text and multimodal perturbations. FIT is noticeably more relatively robust on video perturbations likely due to the use of ViT as the visual encoder. COOT is noticeably low at text robustness, likely due to using pre-extracted text features instead of training a text encoder.

MSRVTT	Video		Text		Video+Text	
	γ^a	γ^r	γ^a	γ^r	γ^a	γ^r
FIT (scratch)	0.93±0.08	0.84±0.18	0.89±0.11	0.75±0.25	0.84±0.10	0.65±0.22
VideoClip (scratch)	0.83±0.15	0.63±0.32	0.89±0.11	0.75±0.24	0.77±0.11	0.50±0.24
FIT (zs)	**0.96±0.06**	**0.91±0.15**	0.92±0.11	**0.81±0.26**	0.89±0.10	**0.73±0.24**
MIL NCE (zs)	0.89±0.08	0.60±0.29	0.94±0.05	0.76±0.20	0.87±0.06	0.51±0.23
UniVL (zs)	0.94±0.05	0.67±0.30	**0.95±0.05**	0.71±0.28	**0.92±0.04**	0.54±0.24
VideoClip (zs)	0.92±0.07	0.66±0.32	0.94±0.06	0.72±0.26	0.88±0.06	0.49±0.25
FIT (ft)	0.92±0.09	0.86±0.15	0.87±0.14	0.77±0.24	0.82±0.13	0.67±0.23
UniVL (ft)	0.82±0.15	0.65±0.29	0.88±0.12	0.77±0.23	0.77±0.12	0.54±0.23
VideoClip (ft)	0.82±0.16	0.66±0.30	0.89±0.12	0.80±0.23	0.76±0.13	0.54±0.24
YouCook2	Video		Text		Video+Text	
	γ^a	γ^r	γ^a	γ^r	γ^a	γ^r
COOT (scratch)	0.79±0.16	0.52±0.36	0.77±0.17	0.49±0.39	0.68±0.11	0.28±0.25
VideoClip (scratch)	0.86±0.11	0.53±0.35	0.91±0.08	0.69±0.28	0.78±0.07	0.27±0.22
MIL NCE (zs)	0.84±0.13	0.53±0.37	0.92±0.09	0.76±0.26	0.80±0.10	**0.42±0.29**
UniVL (zs)	**0.91±0.07**	0.50±0.36	**0.95±0.06**	0.73±0.31	**0.87±0.05**	0.31±0.26
VideoClip (zs)	0.74±0.19	0.50±0.37	0.86±0.13	0.72±0.25	0.65±0.13	0.32±0.25
UniVL (ft)	0.80±0.16	**0.55±0.36**	0.88±0.11	0.72±0.25	0.70±0.10	0.31±0.23
VideoClip (ft)	0.72±0.23	**0.55±0.37**	0.86±0.16	**0.77±0.26**	0.58±0.16	0.33±0.26

Table 6: Average Relative robustness scores and their standard deviations $\gamma^r \pm \sigma$ for text categories on the MSRVTT-QA for the Videoclip model and its training variations.

VideoQA	AddText	Bias	ChangeChar	DropText	Positional	SwapText	TextStyle
scratch	**0.99±0.01**	0.98±0.02	**0.96±0.02**	**0.77±0.23**	**0.97±0.03**	0.94±0.07	**1.0±0.0**
zeroshot	0.97±0.03	0.98±0.01	0.88±0.07	0.67±0.26	0.9±0.06	0.89±0.09	0.98±0.01
finetune	**0.99±0.01**	**0.99±0.01**	**0.96±0.02**	0.75±0.25	0.96±0.03	**0.95±0.06**	0.99±0.01

Table 7: Average Relative robustness scores and their standard deviations $\gamma^r \pm \sigma$ for visual categories on the MSRVTT-QA for the Videoclip model and its training variations.

VideoQA	Blur	Camera	Digital	Noise	Temporal
scratch	0.75±0.14	0.89±0.08	0.71±0.16	0.46±0.2	0.93±0.03
zeroshot	**0.8±0.15**	**0.96±0.07**	**0.8±0.12**	**0.51±0.18**	**1.0±0.01**
finetune	0.78±0.14	0.91±0.08	0.79±0.14	0.49±0.2	0.95±0.02

Table 8: Distribution Shift evaluation using average Relative robustness scores and their standard deviations $\gamma^r \pm \sigma$ and Average Absolute robustness scores and their standard deviations $\gamma^a \pm \sigma$ on MSRVTT and YouCook2 captions respectively when over Natural vs. Machine vs. Artificial (Positional and DropText).

MSRVTT	Natural		Machine		Synthetic	
	γ^a	γ^r	γ^a	γ^r	γ^a	γ^r
FIT (scratch)	0.90±0.09	0.78±0.20	0.94±0.05	0.87±0.11	0.82±0.14	0.61±0.31
VideoClip (scratch)	0.90±0.10	0.77±0.21	0.94±0.04	0.87±0.10	0.82±0.14	0.60±0.30
FIT (zs)	0.93±0.10	**0.84±0.23**	**0.97±0.05**	**0.92±0.13**	0.86±0.14	0.67±0.33
MIL NCE (zs)	0.93±0.04	0.73±0.15	0.96±0.02	0.85±0.09	0.91±0.07	0.66±0.27
UniVL (zs)	**0.97±0.03**	0.80±0.18	**0.97±0.02**	0.85±0.14	**0.92±0.05**	0.50±0.30
VideoClip (zs)	0.95±0.04	0.78±0.16	**0.97±0.03**	0.86±0.14	0.89±0.06	0.52±0.28
FIT (ft)	0.88±0.11	0.79±0.21	0.93±0.06	0.88±0.10	0.80±0.17	0.63±0.31
UniVL (ft)	0.89±0.09	0.79±0.18	0.94±0.05	0.88±0.09	0.81±0.15	0.63±0.29
VideoClip (ft)	0.90±0.10	0.80±0.18	0.94±0.05	0.89±0.09	0.83±0.16	**0.68±0.30**
YouCook2	γ^a	γ^r	γ^a	γ^r	γ^a	γ^r
COOT (scratch)	0.90±0.10	**0.76±0.24**	0.75±0.19	0.44±0.44	0.76±0.16	0.45±0.37
VideoClip (scratch)	0.91±0.07	0.70±0.23	0.95±0.05	0.83±0.18	0.86±0.09	0.52±0.30
MIL NCE (zs)	0.91±0.08	0.74±0.23	0.95±0.05	0.86±0.15	0.89±0.11	**0.67±0.32**
UniVL (zs)	**0.95±0.04**	0.73±0.21	**0.98±0.03**	0.88±0.17	**0.93±0.07**	0.59±0.37
VideoClip (zs)	0.87±0.09	0.75±0.18	0.93±0.06	0.86±0.11	0.79±0.15	0.59±0.29
UniVL (ft)	0.89±0.08	0.75±0.19	0.93±0.05	0.85±0.12	0.82±0.13	0.59±0.30
VideoClip (ft)	0.85±0.12	**0.76±0.19**	0.95±0.06	**0.91±0.10**	0.78±0.20	0.65±0.32

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 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*, pages 1728–1738, 2021.

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In H Larochelle, M Ranzato, R Hadsell, M F Balcan, and H Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[5] Isak Czeresnia Etinger and Alan W Black. Formality style transfer for noisy, user-generated conversations: Extracting labeled, parallel data from unlabeled corpora. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 11–16, Hong Kong, China, November 2019. Association for Computational Linguistics.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA, 1998.

[8] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. COOT: Cooperative Hierarchical Transformer for Video-Text Representation Learning. In H Larochelle, M Ranzato, R Hadsell, M F Balcan, and H Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22605–22618. Curran Associates, Inc., 2020.

[9] Arrow Luo. Video feature extractor. `https://github.com/ArrowLuo/VideoFeatureExtractor/`, 2022.

[10] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation, 2020.

[11] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips, 2019.

[12] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[15] Facebook Research. fairseq. `https://github.com/facebookresearch/fairseq/tree/main/examples/MMPT/scripts/video_feature_extractor`, 2022.

[16] Garrett Reynolds. Gender bender. `https://github.com/Garrett-R/gender_bender`, 2022.

[17] Ganchao Tan, Daqing Liu, Wang Meng, and Zheng-Jun Zha. Learning to discretely compose reasoning module networks for video captioning. *IJCAI-PRICAI*, 2020.

[18] Suramya Tomar. Converting video formats with ffmpeg. *Linux Journal*, 2006(146):10, 2006.

[19] Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, et al. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online, aug 2021. Association for Computational Linguistics.

[20] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. VideoCLIP: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[21] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598, 2018.