Forensics-Bench: A Comprehensive Forgery Detection Benchmark Suite for Large Vision Language Models

Supplementary Material

6. Abbreviations for Forensics-Bench

The detailed abbreviations utilized throughout the paper are listed in Table 5.

Abbreviation	Full Term	Abbreviation	Full Term
		Forgery Semantics	
HS	Human Subject	GS	General Subject
		Forgery Modalities	
RGB	RGB Images	NIR	Near-infrared Images
VID	Videos	RGB&TXT	RGB Images and Texts
		Forgery Tasks	
BC	Forgery Binary Classification	SLD	Forgery Spatial Localization (Detection)
SLS	Forgery Spatial Localization (Segmentation)	TL	Forgery Temporal Localization
		Forgery Types	
ES	Entire Synthesis	SPF	Spoofing
FE	Face Editing	FE&FT	Face Editing & Face Transfer
FE&TAM	Face Editing & Text Attribute Manipulation	FE&TS	Face Editing & Text Swap
FR	Face Reenactment	FSM	Face Swap (Multiple Faces)
FSS	Face Swap (Single Face)	FSS&FE	Face Swap (Single Face) & Face Editing
FSS&TAM	Face Swap (Single Face) & Text Attribute Manipulation	FSS&TS	Face Swap (Single Face) & Text Swap
FT	Face Transfer	CM	Copy-Move
RM	Removal	SPL	Splicing
IE	Image Enhancement	REAL	Real media without being forged
OOC	Out-of-Context	ST	Style Translation
TAM	Text Attribute Manipulation	TS	Text Swap
		Forgery Models	
3D	3D masks	RNN	Recurrent Neural Networks
TR	Transformer	DC	Decoder
DF	Diffusion models	ED	Encoder-Decoder
ED&RNN&GR	Encoder-Decoder&Recurrent Neural Networks&Graphics-based methods	ED&TR	Encoder-Decoder&Transformer
ED&RT	Encoder-Decoder&Retrieval-based methods	ED&GR	Encoder-Decoder&Graphics-based methods
GAN	Generative Adversarial Networks	GAN&TR	Generative Adversarial Networks&Transformer
GAN&RT	Generative Adversarial Networks&Retrieval-based methods	PC	Paper-Cut
Real	Real media without being forged	PR	Print
PRO	Proprietary	RP	Replay
RT	Retrieval-based methods	AR	Auto-regressive models
GR	Graphics-based methods	WILD	Unknown (in the wild)
VAE	Variational Auto-Encoders		

Table 5. The abbreviations of terms mentioned in Forensics-Bench and their corresponding full terms.

7. Data Structure of Forensics-Bench

In Table 6, Table 7 and Table 8, we present all 112 unique forgery detection types from Forensics-Bench, covering 5 designed perspectives characterizing forgeries. These tables include details on sample number, the specific information of 5 designed perspectives in Forensics-Bench and data sources collected under licenses.

8. Other Details of Forensics-Bench

Uniformed metadata. In our benchmark, we design a uniformed metadata structure to standardize and accelerate the construction process of our data samples. As shown in Table 9, the metadata structure is a dictionary with keys divided into three main categories. The first category contains keys such as the image path, image resolution and data source, describing the vanilla information about the raw data. The second category includes keys demonstrating the detailed information of 5 designed perspectives in our benchmark. The third category includes keys for the transformed Q&A, such as the question template, answer (ground truth) and choice list.

Forgery Task	Forgery Semantic	Forgery Type	Forgery Model	Forgery Modality	Data Sources	Sample Number

Forgery Binary Classification	Human Subject	Entire Synthesis	Generative Adversarial Networks	RGB Images	HiFi-IFDL(StyleGANv2-ada on FFHQ) [30];
					HiFi-IFDL(StyleGANv3 on FFHQ) [30];
					DFFD(ProGAN) [15];
					DFFD(StyleGANv1) [15];
					ForgeryNet(StyleGANv2) [32];
					ForgeryNet(DiscoFaceGAN) [32];
					Fake2M(StyleGANv3 on FFHQ/metface) [59]
						2000

Forgery Binary Classification	Human Subject	Entire Synthesis	Generative Adversarial Networks	Near-infrared Images	ForgeryNIR(ProGAN) [91];
					ForgeryNIR(StyleGAN) [91];
					ForgeryNIR(StyleGAN2) [91]
						1200

Forgery Binary Classification	General Subject	Entire Synthesis	Generative Adversarial Networks	RGB Images	HiFi-IFDL(StyleGANv2-ada on AFHQ) [30];
					HiFi-IFDL(styleGANv3 on AFHQ) [30];
					GenImage(BigGAN on ImageNet classes) [116];
					CNN-spot(ProGAN on LSUN) [89];
					CNN-spot(StyleGANv1/v2 on LSUN) [89];
					CNN-spot(BigGAN on ImageNet) [89];
					Fake2M(StyleGAN3 on AFHQ) [59]
						6000

Forgery Binary Classification	Human Subject	Entire Synthesis	Proprietary	RGB Images	Diff(midjourney) [9]	200

Forgery Binary Classification	Human Subject	Entire Synthesis	Diffusion models	RGB Images	Diff(SDXL) [9];
					Diff(FreeDoM_T) [9];
					Diff(HPS) [9];
					Diff(LoRA) [9];
					Diff(DreamBooth) [9];
					Diff(SDXL Refiner) [9];
					Diff(FreeDoM_I) [9]
						1400

Forgery Binary Classification	General Subject	Entire Synthesis	Diffusion models	Videos	Open-Sora-Plan [41]	100
Forgery Binary Classification	General Subject	Entire Synthesis	Auto-regressive models	Videos	Cogvideo [34]	100

Forgery Binary Classification	General Subject	Entire Synthesis	Diffusion models	RGB Images	HiFi-IFDL(GDM on LSUN) [30];
					HiFi-IFDL(LDM on LSUN) [30];
					HiFi-IFDL(DDPM on LSUN) [30];
					HiFi-IFDL(DDIM on LSUN) [30];
					GenImage(SD V1.4 on ImageNet classes) [116];
					GenImage(SD V1.5 on ImageNet classes) [116];
					GenImage(ADM on ImageNet classes) [116];
					GenImage(GLIDE on ImageNet classes) [116];
					Fake2M(SD V2.1) [59];
					Fake2M(SD V1.5) [59];
					Fake2M(IF V1.0) [59];
					DiffusionForensics(ADM on LSUN) [92];
					DiffusionForensics(DDPM on LSUN) [92];
					DiffusionForensics(iDDPM on LSUN) [92];
					DiffusionForensics(PNDM on LSUN) [92];
					DiffusionForensics(LDM on LSUN) [92];
					DiffusionForensics(SD-v1 on LSUN) [92];
					DiffusionForensics(SD-v2 on LSUN) [92];
					DiffusionForensics(ADM on ImageNet) [92];
					DiffusionForensics(SD-v1 on ImageNet) [92]
						5800

Forgery Binary Classification	General Subject	Entire Synthesis	Proprietary	RGB Images	GenImage(Midjourney on ImageNet classes) [116];
					GenImage(Wukong on ImageNet classes) [116];
					Fake2M(Midjourney crawled in the website) [59]
						600

Forgery Binary Classification	General Subject	Entire Synthesis	Variational Auto-Encoders	RGB Images	GenImage(VQDM on ImageNet classes) [116]; DiffusionForensics(VQ-Diffusion on LSUN) [92]	400
Forgery Binary Classification	General Subject	Entire Synthesis	Auto-regressive models	RGB Images	Fake2M(Cogview)[59]	200
Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Graphics-based methods	Videos	FF++(FaceSwap) [75]	140
Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Graphics-based methods	RGB Images	FF++(FaceSwap) [75]	200

Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Encoder-Decoder	Videos	FF++(FaceShifter) [75];
					FF++(Deepfakes) [75];
					ForgeryNet(DeepFaceLab) [32];
					ForgeryNet(FaceShifter) [32];
					CelebDF-v2(Improved Deepfakes) [46];
					DF-TIMIT(Improved Deepfakes) [39, 76]
						1280

Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Encoder-Decoder	RGB Images	FF++(FaceShifter) [75];
					FF++(Deepfakes) [75];
					ForgeryNet(DeepFaceLab) [32];
					ForgeryNet(FaceShifter) [32];
					CelebDF-v2(Improved Deepfakes) [46];
					DF-TIMIT(Improved Deepfakes) [39, 76]
						1400

Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Variational Auto-Encoders	Videos	DeeperForensics(DeepFake VAE) [36]	200
Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Variational Auto-Encoders	RGB Images	DeeperForensics(DeepFake VAE) [36]	200
Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Recurrent Neural Networks	Videos	ForgeryNet(FSGAN) [32];	200
Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Recurrent Neural Networks	RGB Images	ForgeryNet(FSGAN) [32];	200

Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Unknown (in the wild)	Videos	DFDCP [17];
					WildDeepfake [117];
					DFD [4]
						400

Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Unknown (in the wild)	RGB Images	DFDCP [17];
					WildDeepfake [117];
					DFD [4]
						400

Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Diffusion models	RGB Images	Diff(DiffFace) [9]; Diff(DCFace) [9]	400
Forgery Binary Classification	Human Subject	Face Swap (Multiple Faces)	Encoder-Decoder,Recurrent Neural Networks,Graphics-based methods	Videos	FFIW(DeepFaceLab, FSGAN, FaceSwap) [114]; DF-Platter(FaceShifter) [66]	200
Forgery Binary Classification	Human Subject	Face Swap (Multiple Faces)	Encoder-Decoder,Recurrent Neural Networks,Graphics-based methods	RGB Images	FFIW(DeepFaceLab, FSGAN, FaceSwap) [114]; DF-Platter(FaceShifter) [66]	200
Forgery Binary Classification	Human Subject	Face Transfer	Graphics-based methods	Videos	ForgeryNet(BlendFace) [32]; ForgeryNet(MMReplacement) [32]	300
Forgery Binary Classification	Human Subject	Face Transfer	Graphics-based methods	RGB Images	ForgeryNet(BlendFace) [32]; ForgeryNet(MMReplacement) [32]	400
Forgery Binary Classification	Human Subject	Face Reenactment	Graphics-based methods	Videos	FF++(Face2Face) [75]	140
Forgery Binary Classification	Human Subject	Face Reenactment	Graphics-based methods	RGB Images	FF++(Face2Face) [75]	200
Forgery Binary Classification	Human Subject	Face Reenactment	Encoder-Decoder	Videos	FF++(NeuralTextures)[75]	140

Table 6. Forensics-Bench data structure (part 1): including the detailed information of 5 designed perspectives characterizing forgeries, sample number and data sources collected under licenses.

Details of forgery types. In our benchmark, we roughly classify previous forgeries into 21 types, which are summarized as follows.

- Entire Synthesis: In our benchmark, this refers to forgeries that are synthesized from scratch without a basis on real media. For instance, vanilla GAN models and diffusion models can generate forgeries from random Gaussian noises. Representative datasets of this type include CNN-spot [89] and DiffusionForensics [92].

Forgery Task	Forgery Semantic	Forgery Type	Forgery Model	Forgery Modality	Data Sources	Sample Number
Forgery Binary Classification	Human Subject	Face Reenactment	Encoder-Decoder	RGB Images	FF++(NeuralTextures) [75]; ForgeryNet(FirstOrderMotion) [32]	400
Forgery Binary Classification	Human Subject	Face Reenactment	Recurrent Neural Networks	Videos	ForgeryNet(ATVG-Net) [32]; ForgeryNet(Talking-head Video) [32]	400
Forgery Binary Classification	Human Subject	Face Reenactment	Recurrent Neural Networks	RGB Images	ForgeryNet(ATVG-Net) [32]; ForgeryNet(Talking-head Video) [32]	400

Forgery Binary Classification	Human Subject	Face Editing	Encoder-Decoder	RGB Images	HiFi-IFDL(starGANv2 on CelebaHQ) [30];
					HiFi-IFDL(HiSD on CelebaHQ) [30];
					HiFi-IFDL(STGAN on CelebaHQ) [30];
					DFFD(starGAN on CelebA) [15];
					ForgeryNet(starGANv2) [32];
					ForgeryNet(MaskGAN) [32];
					ForgeryNet(SC-FEGAN) [32];
					CNN-spot(starGAN) [89]
						1400

Forgery Binary Classification	Human Subject	Style Translation	Encoder-Decoder	Near-infrared Images	ForgeryNIR(CycleGAN) [91]	400
Forgery Binary Classification	Human Subject	Face Editing	Proprietary	RGB Images	DFFD(FaceAPP on FFHQ) [15]	200

Forgery Binary Classification	Human Subject	Face Editing	Diffusion models	RGB Images	Diff(Imagic) [9];
					Diff(CoDiff) [9];
					Diff(CycleDiff) [9]
						600

Forgery Binary Classification	General Subject	Style Translation	Encoder-Decoder	RGB Images	CNN-spot(CycleGAN) [89]; CNN-spot(GauGAN)[89]	1260
Forgery Binary Classification	General Subject	Style Translation	Decoder	RGB Images	CNN-spot(CRN) [89]; CNN-spot(IMLE) [89]	400
Forgery Binary Classification	General Subject	Image Enhancement	Encoder-Decoder	RGB Images	CNN-spot(SITD) [89]; CNN-spot(SAN) [89]	380
Forgery Binary Classification	Human Subject	Face Editing,Face Transfer	Encoder-Decoder,Graphics-based methods	RGB Images	ForgeryNet(StarGAN2+BlendFace) [32]	200
Forgery Binary Classification	Human Subject	Face Swap (Single Face),Face Editing	Encoder-Decoder	Videos	ForgeryNet(DeepFaceLab-StarGAN2) [32]	200
Forgery Binary Classification	Human Subject	Face Swap (Single Face),Face Editing	Encoder-Decoder	RGB Images	ForgeryNet(DeepFaceLab-StarGAN2) [32]	200
Forgery Binary Classification	General Subject	Copy&Move	Graphics-based methods	RGB Images	HiFi-IFDL(PSCC-Net) [30]	200
Forgery Binary Classification	General Subject	Removal	Encoder-Decoder	RGB Images	HiFi-IFDL(PSCC-Net) [30]	200
Forgery Binary Classification	General Subject	Splicing	Graphics-based methods	RGB Images	HiFi-IFDL(PSCC-Net) [30]	200
Forgery Binary Classification	Human Subject	Face Swap (Single Face)	Encoder-Decoder	RGB Images,Texts	DGM4(SimSwap) [78]; DGM4(InfoSwap) [78]	400
Forgery Binary Classification	Human Subject	Face Editing	Encoder-Decoder	RGB Images,Texts	DGM4(HFGI) [78]	200
Forgery Binary Classification	Human Subject	Face Editing	Generative Adversarial Networks	RGB Images,Texts	DGM4(StyleCLIP) [78]	200
Forgery Binary Classification	Human Subject	Text Swap	Retrieval-based methods	RGB Images,Texts	DGM4(retrieval) [78]	200
Forgery Binary Classification	Human Subject	Text Attribute Manipulation	Transformer	RGB Images,Texts	DGM4(B-GST) [78]	200
Forgery Binary Classification	Human Subject	Face Swap (Single Face),Text Swap	Encoder-Decoder,Retrieval-based methods	RGB Images,Texts	DGM4(SimSwap+retrieval) [78]; DGM4(InfoSwap+retrieval) [78]	400
Forgery Binary Classification	Human Subject	Face Editing,Text Swap	Encoder-Decoder,Retrieval-based methods	RGB Images,Texts	DGM4(HFGI+retrieval) [78]	200
Forgery Binary Classification	Human Subject	Face Editing,Text Swap	Generative Adversarial Networks,Retrieval-based methods	RGB Images,Texts	DGM4(StyleCLIP+retrieval) [78]	200
Forgery Binary Classification	Human Subject	Face Swap (Single Face),Text Attribute Manipulation	Encoder-Decoder,Transformer	RGB Images,Texts	DGM4(SimSwap+B-GST) [78]; DGM4(InfoSwap+B-GST) [78]	400
Forgery Binary Classification	Human Subject	Face Editing,Text Attribute Manipulation	Encoder-Decoder,Transformer	RGB Images,Texts	DGM4(HFGI+B-GST) [78]	200
Forgery Binary Classification	Human Subject	Face Editing,Text Attribute Manipulation	Generative Adversarial Networks,Transformer	RGB Images,Texts	DGM4(StyleCLIP+B-GST) [78]	200
Forgery Binary Classification	Human Subject	Out-of-Context	Retrieval-based methods	RGB Images,Texts	NewsCLIPpings [60]	100
Forgery Binary Classification	Human Subject	Face Spoofing	Print	RGB Images	CelebA-Spoof [109]	200
Forgery Binary Classification	Human Subject	Face Spoofing	Paper Cut	RGB Images	CelebA-Spoof [109]	200
Forgery Binary Classification	Human Subject	Face Spoofing	Replay	RGB Images	CelebA-Spoof [109]	200
Forgery Binary Classification	Human Subject	Face Spoofing	3D masks	RGB Images	CelebA-Spoof [109]	200

Forgery Spatial Localization (Segmentation)	Human Subject	Face Swap (Single Face)	Encoder-Decoder	Videos	HiFi-IFDL(FaceShifter on Youtube video) [30];
					DFFD(DeepFaceLab) [15];
					DFFD(Deepfakes) [15];
					ForgeryNet(FaceShifter) [32];
					ForgeryNet(DeepFaceLab) [32]
						309

Forgery Spatial Localization (Segmentation)	Human Subject	Face Swap (Single Face)	Encoder-Decoder	RGB Images	HiFi-IFDL(FaceShifter on Youtube video) [30];
					DFFD(DeepFaceLab) [15];
					DFFD(Deepfakes) [15];
					ForgeryNet(FaceShifter) [32];
					ForgeryNet(DeepFaceLab) [32]
						598

Forgery Spatial Localization (Segmentation)	Human Subject	Face Swap (Single Face)	Graphics-based methods	Videos	FF++(FaceSwap) [75]	140
Forgery Spatial Localization (Segmentation)	Human Subject	Face Swap (Single Face)	Graphics-based methods	RGB Images	FF++(FaceSwap) [75]	200
Forgery Spatial Localization (Segmentation)	Human Subject	Face Swap (Single Face)	Recurrent Neural Networks	RGB Images	ForgeryNet(FSGAN) [32]	200
Forgery Spatial Localization (Segmentation)	Human Subject	Face Transfer	Graphics-based methods	Videos	ForgeryNet(BlendFace) [32]; ForgeryNet(MMReplacement) [32]	231
Forgery Spatial Localization (Segmentation)	Human Subject	Face Transfer	Graphics-based methods	RGB Images	ForgeryNet(BlendFace) [32]; ForgeryNet(MMReplacement) [32]	400
Forgery Spatial Localization (Segmentation)	Human Subject	Face Reenactment	Graphics-based methods	Videos	FF++(Face2Face) [75]	140

Table 7. Forensics-Bench data structure (part 2): including the detailed information of 5 designed perspectives characterizing forgeries, sample number and data sources collected under licenses.

- Spoofing: In our benchmark, this refers to forgeries that present a fake version of a legitimate user's face to bypass authentication, such as the printed photograph of a user's face, a recorded video of the target user and 3D masks that mimic the target's facial structures. Representative datasets of this type include CelebA-Spoof [109].
- Face Editing: In our benchmark, this refers to forgeries that modify the external attributes of human faces, such as facial hair, age and gender. Representative datasets of this type include ForgeryNet [32].
- Face Swap (Single Face): In our benchmark, this refers to forgeries that exchange one person's facial features with another, changing the original identity of the depicted person. Representative datasets of this type include CelebDF-v2 [46].
- Face Swap (Multiple Faces): In our benchmark, this refers to forgeries that exchange more than one person's facial features with other human faces in one media. Representative datasets of this type include OpenForensics [42].
- Face Transfer: In our benchmark, this refers to forgeries that transfer both the identity-aware and identity-agnostic content (such as the pose and expression) of the source face to the target face. This follows the design proposed in ForgeryNet [32].
- Face Reenactment: In our benchmark, this refers to forgeries that transfer the facial expressions, movements, and emotions of one person's face to another person's face. Representative datasets of this type include FF++ [75].
- Copy-Move: In our benchmark, this refers to forgeries that involve copying a portion of an image and pasting it elsewhere within the same image. Representative datasets of this type include HiFi-IFDL [30].
- Removal: In our benchmark, this refers to forgeries that involve removing an object or region from an image and filling in the removed area to maintain the visual coherence, which is also known as "inpainting". Representative datasets of this type include HiFi-IFDL [30].

Forgery Task	Forgery Semantic	Forgery Type	Forgery Model	Forgery Modality	Data Sources	Sample Number
Forgery Spatial Localization (Segmentation)	Human Subject	Face Reenactment	Graphics-based methods	RGB Images	FF++(Face2Face) [75]	200
Forgery Spatial Localization (Segmentation)	Human Subject	Face Reenactment	Encoder-Decoder	RGB Images	ForgeryNet(FirstOrderMotion) [32]	200
Forgery Spatial Localization (Segmentation)	Human Subject	Face Reenactment	Recurrent Neural Networks	RGB Images	ForgeryNet(ATVG-Net) [32]; ForgeryNet(Talking-head Video) [32]	400

Forgery Spatial Localization (Segmentation)	Human Subject	Face Editing	Encoder-Decoder	RGB Images	HiFi-IFDL(STGAN on CelebaHQ) [30];
					DFFD(starGAN on CelebA) [15];
					ForgeryNet(starGANv2) [32];
					ForgeryNet(MaskGAN) [32];
					ForgeryNet(SC-FEGAN) [32]
						800

Forgery Spatial Localization (Segmentation)	Human Subject	Face Editing	Proprietary	RGB Images	DFFD(FaceAPP on FFHQ) [15]	200
Forgery Spatial Localization (Segmentation)	Human Subject	Face Editing,Face Transfer	Encoder-Decoder,Graphics-based methods	RGB Images	ForgeryNet(StarGAN2+BlendFace) [32]	200
Forgery Spatial Localization (Segmentation)	Human Subject	Face Swap (Single Face),Face Editing	Encoder-Decoder	Videos	ForgeryNet(DeepFaceLab-StarGAN2) [32]	100
Forgery Spatial Localization (Segmentation)	Human Subject	Face Swap (Single Face),Face Editing	Encoder-Decoder	RGB Images	ForgeryNet(DeepFaceLab-StarGAN2) [32]	200
Forgery Spatial Localization (Segmentation)	General Subject	Copy&Move	Graphics-based methods	RGB Images	HiFi-IFDL(PSCC-Net) [30]	200
Forgery Spatial Localization (Segmentation)	General Subject	Removal	Encoder-Decoder	RGB Images	HiFi-IFDL(PSCC-Net) [30]	200
Forgery Spatial Localization (Segmentation)	General Subject	Splicing	Graphics-based methods	RGB Images	HiFi-IFDL(PSCC-Net) [30]	200

Forgery Spatial Localization (Segmentation)	Human Subject	Entire Synthesis	Generative Adversarial Networks	RGB Images	DFFD(ProGAN) [15];
					DFFD(StyleGANv1) [15];
					ForgeryNet(StyleGANv2) [32];
					ForgeryNet(DiscoFaceGAN) [32]
						800

Forgery Spatial Localization (Segmentation)	Human Subject	Face Swap (Multiple Faces)	Generative Adversarial Networks	RGB Images	OpenForensics [42]	200
Forgery Spatial Localization (Detection)	Human Subject	Face Swap (Multiple Faces)	Generative Adversarial Networks	RGB Images	OpenForensics [42]	200
Forgery Spatial Localization (Segmentation)	Human Subject	Face Swap (Multiple Faces)	Graphics-based methods,Recurrent Neural Networks,Encoder-Decoder	Videos	FFIW(DeepFaceLab, FSGAN, FaceSwap) [114]	200
Forgery Spatial Localization (Segmentation)	Human Subject	Face Swap (Multiple Faces)	Graphics-based methods,Recurrent Neural Networks,Encoder-Decoder	RGB Images	FFIW(DeepFaceLab, FSGAN, FaceSwap) [114]	200
Forgery Spatial Localization (Detection)	Human Subject	Face Swap (Single Face)	Encoder-Decoder	RGB Images,Texts	DGM4(SimSwap) [78]; DGM4(InfoSwap) [78]	400
Forgery Spatial Localization (Detection)	Human Subject	Face Editing	Encoder-Decoder	RGB Images,Texts	DGM4(HFGI) [78]	200
Forgery Spatial Localization (Detection)	Human Subject	Face Editing	Generative Adversarial Networks	RGB Images,Texts	DGM4(StyleCLIP) [78]	200
Forgery Spatial Localization (Detection)	Human Subject	Text Swap	Retrieval-based methods	RGB Images,Texts	DGM4(retrieval) [78]	200
Forgery Spatial Localization (Detection)	Human Subject	Text Attribute Manipulation	Transformer	RGB Images,Texts	DGM4(B-GST) [78]	200
Forgery Spatial Localization (Detection)	Human Subject	Face Swap (Single Face),Text Swap	Encoder-Decoder,Retrieval-based methods	RGB Images,Texts	DGM4(SimSwap+retrieval) [78]; DGM4(InfoSwap+retrieval) [78]	400
Forgery Spatial Localization (Detection)	Human Subject	Face Editing,Text Swap	Encoder-Decoder,Retrieval-based methods	RGB Images,Texts	DGM4(HFGI+retrieval) [78]	200
Forgery Spatial Localization (Detection)	Human Subject	Face Editing,Text Swap	Generative Adversarial Networks,Retrieval-based methods	RGB Images,Texts	DGM4(StyleCLIP+retrieval) [78]	200
Forgery Spatial Localization (Detection)	Human Subject	Face Swap (Single Face),Text Attribute Manipulation	Encoder-Decoder,Transformer	RGB Images,Texts	DGM4(SimSwap+B-GST) [78]; DGM4(InfoSwap+B-GST) [78]	400
Forgery Spatial Localization (Detection)	Human Subject	Face Editing,Text Attribute Manipulation	Encoder-Decoder,Transformer	RGB Images,Texts	DGM4(HFGI+B-GST) [78]	200
Forgery Spatial Localization (Detection)	Human Subject	Face Editing,Text Attribute Manipulation	Generative Adversarial Networks,Transformer	RGB Images,Texts	DGM4(StyleCLIP+B-GST) [78]	200
Forgery Temporal Localization	Human Subject	Face Swap (Single Face)	Encoder-Decoder	Videos	ForgeryNet(DeepFaceLab) [32]; ForgeryNet(FaceShifter) [32]	400
Forgery Temporal Localization	Human Subject	Face Swap (Single Face)	Recurrent Neural Networks	Videos	ForgeryNet(FSGAN) [32]	200
Forgery Temporal Localization	Human Subject	Face Transfer	Graphics-based methods	Videos	ForgeryNet(BlendFace) [32]; ForgeryNet(MMReplacement) [32]	300
Forgery Temporal Localization	Human Subject	Face Reenactment	Recurrent Neural Networks	Videos	ForgeryNet(ATVG-Net) [32]; ForgeryNet(Talking-head Video) [32]	400
Forgery Temporal Localization	Human Subject	Face Swap (Single Face),Face Editing	Encoder-Decoder	Videos	ForgeryNet(DeepFaceLab-StarGAN2) [32]	200
Forgery Binary Classification	Human Subject	Real	Real	RGB Images,Texts	DGM4 [78]	2000

Forgery Binary Classification	Human Subject	Real	Real	RGB Images	DFFD(FFHQ) [15];
					DiffusionForensics(CelebAHQ) [92];
					DeeperForensics [36];
					FF++ [75];
					CelebDF-v2 [46];
					FFIW [114];
					CelebA-Spoof [109]
						4000

Forgery Binary Classification	General Subject	Real	Real	RGB Images	CNN-spot [89];
					DiffusionForensics(LSUN, ImageNet) [92];
					COCO2017val [51]
						4000

Forgery Binary Classification	Human Subject	Real	Real	Videos	FF++ [75];
					CelebDF-v2 [46];
					DeeperForensics [36];
					FFIW [114];
					CelebA-Spoof [109]
						178

Forgery Spatial Localization (Segmentation)	Human Subject	Real	Real	RGB Images	DFFD(FFHQ) [15];
					DiffusionForensics(CelebAHQ) [92];
					DeeperForensics [36];
					FF++ [75];
					CelebDF-v2 [46];
					FFIW [114];
					CelebA-Spoof [109]
						1600

Forgery Spatial Localization (Segmentation)	General Subject	Real	Real	RGB Images	CNN-spot [89]; DiffusionForensics(LSUN, ImageNet) [92] COCO2017val [51]	1500

Forgery Spatial Localization (Segmentation)	Human Subject	Real	Real	Videos	FF++ [75];
					CelebDF-v2 [46];
					DeeperForensics [36];
					FFIW [114]
						178

Forgery Spatial Localization (Detection)	Human Subject	Real	Real	RGB Images,Texts	DGM4 [78]	1000

Forgery Spatial Localization (Detection)	Human Subject	Real	Real	RGB Images	DFFD(FFHQ) [15];
					DiffusionForensics(CelebAHQ) [92];
					DeeperForensics [36];
					FF++ [75];
					CelebDF-v2 [46];
					FFIW [114];
					CelebA-Spoof [109]
						1100

Forgery Spatial Localization (Detection)	General Subject	Real	Real	RGB Images	CNN-spot [89]; DiffusionForensics(LSUN, ImageNet) [92]	1000
Forgery Temporal Localization	Human Subject	Real	Real	Videos	ForgeryNet [32]	378

Table 8. Forensics-Bench data structure (part 3): including the detailed information of 5 designed perspectives characterizing forgeries, sample number and data sources collected under licenses.

- Splicing: In our benchmark, this refers to forgeries that involve combining elements from two or more different images to create a composite image. Representative datasets of this type include HiFi-IFDL [30].
- Image Enhancement: In our benchmark, this refers to forgeries where enhancements are deliberately applied to alter the appearance of an image, such as image super-resolution and low-light imaging. Representative datasets of this type include CNN-spot [89].
- Out-of-Context: In our benchmark, this refers to forgeries where the presentation of an authentic image, video, or media clip is repurposed with a misleading or deceptive text. Representative datasets of this type include NewsCLIPpings [60].
- Style Translation: In our benchmark, this refers to forgeries which transform the visual style of one image while preserving the content of another image. Representative datasets of this type include CNN-spot [89].
- Text Attribute Manipulation: In our benchmark, this refers to forgeries that alter the sentiment tendency of a given text while preserving its core content or meaning. This follows the design in DGM4 [78].

Keys	Example 1	Example 2
Image Path	/path/to/image	/path/to/image
Image Resolution	299x299	1280x720
Data Source	DFFD_StyleGANv1_ffhq	ForgeryNet_12_seg
Forgery Semantic	Human	Human
Forgery Modality	RGB Image	RGB Image
Forgery Task	Forgery Binary Classification	Forgery Spatial Localization (Segmentation)
Forgery Type	Entire Synthesis	Face Editing
Forgery Model	Generative Adversarial Networks	Encoder-Decoder
Question Template	What is the authenticity of the image?	Which segmentation map denotes the forged area in the image most accurately?
Choice List	[AI-generated, non AI-generated]	[Candidate 1, Candidate 2, Candidate 3, Candidate 4]
Answer	AI-generated	Candidate 4

Table 9. Examples of the uniformed metadata.

- Text Swap: In our benchmark, this refers to forgeries that alter the overall semantic of a text with word substitution while preserving words regarding the main character. This follows the design in DGM4 [78].
- Face Editing & Text Attribute Manipulation: In our benchmark, this refers to forgeries that are produced under the combination of both Face Editing & Text Attribute Manipulation. This follows the design in DGM4 [78].
- Face Editing & Text Swap: In our benchmark, this refers to forgeries that are produced under the combination of both Face Editing & Text Swap. This follows the design in DGM4 [78].
- Face Editing & Face Transfer: In our benchmark, this refers to forgeries that are produced under the combination of both Face Editing & Face Transfer. This follows the design in ForgeryNet [32].
- Face Swap (Single Face) & Face Editing: In our benchmark, this refers to forgeries that are produced under the combination of both Face Swap (Single Face) & Face Editing. This follows the design in ForgeryNet [32].
- Face Swap (Single Face) & Text Attribute Manipulation: In our benchmark, this refers to forgeries that are produced under the combination of both Face Swap (Single Face) & Text Attribute Manipulation. This follows the design in DGM4 [78].
- Face Swap (Single Face) & Text Swap: In our benchmark, this refers to forgeries that are produced under the combination of both Face Swap (Single Face) & Text Swap. This follows the design in DGM4 [78].

Details of forgery models. In our benchmark, we roughly divide previous forgeries into 22 categories from the perspective of forgery model. We summarize the details as follows.

- Generative Adversarial Networks: In our benchmark, this refers to forgeries that are generated with vanilla GANs, namely a pair of adversarially trained generator and discriminator. Representative datasets of this category include CNN-spot [89].
- Diffusion models: In our benchmark, this refers to forgeries that are generated with vanilla diffusion models, such as DDPM [33]. Representative datasets of this category include DiffusionForensics [92].
- Encoder-Decoder: In our benchmark, this represents forgery models which commonly take real media as input, and are typically used to separate the identity information from identity-agnostic attributes, then alter or exchange the facial representations. This kind of models usually features an encoder-decoder structure. This follows the design in ForgeryNet [32] and representative datasets of this category include CelebDF-v2 [46] and FF++ [75].
- Recurrent Neural Networks: In our benchmark, this represents forgery models which are commonly used to alter sequential and dynamic-length data like videos. This follows the design in ForgeryNet [32].
- Proprietary: In our benchmark, this represents closed-source forgery models commonly used for commercial purposes, like Midjourney. Representative datasets of this category include GenImage [116].
- 3D masks: In our benchmark, this represents forgeries which are produced based on 3D masks designed to look like real users, commonly used for face spoofing. Representative datasets of this category include CelebA-Spoof [109].
- Print: In our benchmark, this represents forgeries which are produced based on a printed photograph of a face, in order to trick facial recognition systems. Representative datasets of this category include CelebA-Spoof [109].
- Paper-Cut: In our benchmark, this represents forgeries which are produced based on a printed photograph of a face with specific modifications, such as eye and mouth cutouts. Representative datasets of this kind include CelebA-Spoof [109].
- Replay: In our benchmark, this represents forgeries which are produced by displaying a recorded video or image sequence of the face on a screen. Representative datasets of this category include CelebA-Spoof [109].
- Transformer: In our benchmark, this represents forgery models that are mainly used to modify texts, such as altering the sentiment tendency. Representative datasets of this category include DGM4 [78].
- Decoder: In our benchmark, this represents forgery models which are mainly used to perform style translations, commonly featuring a decode-only structure. Representative datasets of this category include CNN-spot [89].
- Graphics-based methods: In our benchmark, this represents forgeries that are mainly produced with traditional graphics

formations. This follows the design in ForgeryNet [32].

- Retrieval-based methods: In our benchmark, this represents forgeries that are produced by retrieving existing data. Representative datasets of this category include DGM4 [78].
- Unknown (in the wild): In our benchmark, this represents forgeries with unknown sources. Representative datasets of this category include DFPCP [17].
- Variational Auto-Encoders: In our benchmark, this represents forgeries that are generated with typical Variational Auto-Encoders. Representative datasets of this category include DeeperForensics [36].
- Auto-regressive models: In our benchmark, this represents forgery models which are commonly used to generate video data with no basis of real media, such as CogVideo [34].
- Encoder-Decoder&Retrieval-based methods: In our benchmark, this represents forgeries that are produced under the combination of Encoder-Decoder&Retrieval-based methods. This follows the design in DGM4 [78].
- Encoder-Decoder&Recurrent Neural Networks&Graphics-based methods: In our benchmark, this represents forgeries that are produced under the combination of Encoder-Decoder&Recurrent Neural Networks&Graphics-based methods. Representative datasets of this category include FFIW [114].
- Generative Adversarial Networks&Retrieval-based methods: In our benchmark, this represents forgeries that are produced by the combination of Generative Adversarial Networks&Retrieval-based methods. This follows the design in DGM4 [78].
- Encoder-Decoder&Transformer: In our benchmark, this represents forgeries that are produced under the combination of Encoder-Decoder&Transformer. This follows the design in DGM4 [78].
- Generative Adversarial Networks&Transformer: In our benchmark, this represents forgeries that are produced under the combination of Generative Adversarial Networks&Transformer. This follows the design in DGM4 [78].
- Encoder-Decoder&Graphics-based methods: In our benchmark, this represents forgeries that are produced under the combination of Encoder-Decoder&Graphics-based methods. This follows the design in ForgeryNet [32].

Details of forgery tasks. In our benchmark, we roughly divide previous forgeries into 4 categories from the perspective of forgery task. We summarize the details as follows.

- Forgery Binary Classification: This task aims to identify whether a given input (image, video, or text) is genuine or fake (forged). For instance, we can design the question template as *What is the authenticity of the image?* with two choice selections *AI-generated* and *non AI-generated* for HS-RGB-BC-ES-GAN (Please refer to Table 5 for the full term).
- Forgery Spatial Localization (Detection): This task aims to determine the specific regions within the input that have been altered, tampered with, or manipulated. For instance, we can design the question template as *Please detect the forged area in this image and the forged text in the corresponding caption: "Gen Prayuth Chanocha says democracy will only return after reforms are put in place". The output format for the forged area should be a list of bounding boxes, namely [x, y, w, h], representing the coordinates of the top-left corner of the bounding box, as well as the width and height of the bounding box. The width of the input image is 624 and the height is 351. The output format for the forged text should be the a list of token positions in the whole caption, where the initial position index starts from 0. The token length of the input caption is 14.*. The corresponding choice list is: *A.{ "forged area": [[274, 46, 358, 167]], "forged text": [] }, B. { "forged area": [[274, 46, 358, 167], [220, 35, 330, 169]], "forged text": [5] }, C. { "forged area": [[274, 46, 358, 167], [186, 122, 333, 196]], "forged text": [1, 6] }, D. { "forged area": [[274, 46, 332, 141], [1, 120, 295, 192]], "forged text": [] }*. This example is for HS-RGB&TXT-SLD-FE-ED (Please refer to Table 5 for the full term).
- Forgery Spatial Localization (Segmentation): This task aims to precisely outline the regions of tampered or manipulated content within the digital media using pixel-wise classification. For instance, we can design the question template as *Which segmentation map denotes the forged area in the image most accurately?* with four choice selections *[Candidate 1, Candidate 2, Candidate 3, Candidate 4]*, each of which points to a segmentation map. This example is for HS-RGB-SLS-FE-ED (Please refer to Table 5 for the full term).
- Forgery Temporal Localization: This task aims to detect the tampered or manipulated segments within a video. For instance, we can design the question template as *Please locate the forged frames in the given set of frames, which are sampled from a video. The output format should be the a list of indexes indicating the forged frames. The initial index starts from 0.*. The corresponding choice list is: *A. [0, 1, 5], B. [1], C. [0], D. [0, 1]*. This example is for HS-VID-TL-FSS-ED (Please refer to Table 5 for the full term).

9. LVLMs Model Details

In this section, we present the summary of the LVLMs utilized in this paper, detailing their parameter sizes, visual encoders, and LLMs, which is shown in Table 10. We follow the evaluation tool [22] provided in OpenCompass [12] for the evaluations.

Models	Parameters	Vision Encoder	LLM
GPT4o [69]	-	-	-
Gemini1.5 ProVision [84]	-	-	-
Claude3.5-Sonnet [1]	-	-	-
LLaVA-Next-34B [54]	34.8B	CLIP ViT-L/14	Nous-Hermes-2-Yi-34B
LLaVA-v1.5-7B-XTuner [13]	7.2B	CLIP ViT-L/14	Vicuna-v1.5-7B
LLaVA-v1.5-13B-XTuner [13]	13.4B	CLIP ViT-L/14	Vicuna-v1.5-13B
InternVL-Chat-V1-2 [8, 86]	40B	InternViT-6B	Nous-Hermes-2-Yi-34B
LLaVA-NEXT-13B [54]	13.4B	CLIP ViT-L/14	Vicuna-v1.5-13B
mPLUG-Owl2 [101]	8.2B	CLIP ViT-L/14	LLaMA2-7B
LLaVA-v1.5-7B [52, 53]	7.2B	CLIP ViT-L/14	Vicuna-v1.5-7B
LLaVA-v1.5-13B [52, 53]	13.4B	CLIP ViT-L/14	Vicuna-v1.5-13B
Yi-VL-34B [103]	34.6B	CLIP ViT-H/14	Nous-Hermes-2-Yi-34B
CogVLM-Chat [90]	17B	EVA-CLIP-E	Vicuna-v1.5-7B
XComposer2 [21]	7B	CLIP ViT-L/14	InternLM2-7B
LLaVA-InternLM2-7B [13]	8.1B	CLIP ViT-L/14	InternLM2-7B
VisualGLM-6B	8B	EVA-CLIP	ChatGLM-6B
LLaVA-NEXT-7B [54]	7.1B	CLIP ViT-L/14	Vicuna-v1.5-7B
LLaVA-InternLM-7B [13]	7.6B	CLIP ViT-L/14	InternLM-7B
ShareGPT4V-7B [7]	7.2B	CLIP ViT-L/14	Vicuna-v1.5-7B
InternVL-Chat-V1-5 [8, 86]	40B	InternViT-6B	Nous-Hermes-2-Yi-34B
DeepSeek-VL-7B [58]	7.3B	SAM-B & SigLIP-L	DeekSeek-7B
Yi-VL-6B [103]	6.6B	CLIP ViT-H/14	Yi-6B
InstructBLIP-13B [14]	13B	EVA-CLIP ViT-G/14	Vicuna-v1.5-13B
Qwen-VL-Chat [2]	9.6B	CLIP ViT-G/16	Qwen-7B
Monkey-Chat [48]	9.8B	CLIP-ViT-BigHuge	Qwen-7B

Table 10. Model architecture of 25 LVLMs evaluated on Forensics-Bench.

10. Additional Experiments

Single-image input *vs* Multi-images input. The ability to process multiple images is essential for large vision language models, which may also facilitate LVLMs to understand forgeries of sequential data like videos. For example, frames of a real video may transition smoothly and naturally, whereas a fake video may exhibit inter-frame inconsistencies. To this end, we propose to analyze the effects of single-image prompt and multi-images prompt on LVLMs with capabilities to understand multiple images. Specifically, we collect the subset of our Forensics-Bench featuring video modality, and feed LVLMs with single-image input and multi-images input. Note that the single-image input is generated by piecing together sampled frames into one big input image, as shown in Figure 1. The results are demonstrated in Table 11, where the evaluated LVLMs also support multiple images as input. We find that LVLMs, like InternVL-Chat-V1-2 and Gemini-1.5-Pro, effectively exploited the relations among frames to perform forgery detections, while other LVLMs faced challenges in extracting meaningful information to determine the authenticity of the input frames, highlighting the unique difficulties of video forgery detections.

Experiments on prompt engineering. In the main paper, we mainly focused on baseline evaluations, following the OpenCompass [12] protocol and using default system prompts recommended by each LVLM, which are already well-trained. Nevertheless, beyond the baseline results, we have conducted experiments, adding a new forgery-related prompt: "*Please make your decision using forgery detection techniques, such as examining facial features, blending artifacts, lighting irregularities, and any other inconsistencies that may indicate manipulations.*". Results in Table 12 show guiding LVLMs to focus on such forgery traces boosted performance to some extent, which may inspire future studies.

More experiments on forgery attribution. In this section, we explore methods to enhance LVLMs' performance on the task of forgery attribution. To this end, we have conducted experiments by adding detailed introductions of different forgery models into the prompt, as detailed in Appendix 8, aiming to reduce LVLMs' potential misunderstandings for forgery attribution. Results in Table 13 show that this improved LVLMs' performance, which may inspire future studies.

Experiments on visual prompt engineering. In this section, we have conducted experiments where we added bounding

Model	InternVL-Chat-V1-2	mPLUG-Owl2	Gemini-1.5-Pro	InternVL-Chat-V1-5	Qwen-VL-Chat	Claude3V-Sonnet
Single-Image Prompt	62.9	59.8	38.8	52.2	38.9	35.9
Multi-Images Prompt	63.9	36.3	40.9	34.8	25.7	30.2

Table 11. The performance comparison between single-image input and multi-images input.

Model	Baseline	+Prompt Engineering
LLaVA-v1.5-7B-XTuner	65.7	**67.6**
LLaVA-v1.5-13B-XTuner	65.2	**67.1**
LLaVA-NEXT-13B	58.0	**61.3**

Table 12. Experiments on prompt engineering.

Model	Baseline	+Detailed Introductions of Forgery Models
LLaVA-NEXT-34B	44.0	**55.7**
InternVL-Chat-V1-2	41.6	**55.6**
LLaVA-v1.5-7B-XTuner	42.2	**49.6**
mPLUG-Owl2	39.9	**45.4**

Table 13. More experiments on forgery attribution.

Model	Baseline	+Prompt Engineering (Visual)
LLaVA-v1.5-7B-XTuner	83.5	**87.6**
LLaVA-NEXT-34B	84.1	**85.7**
InternVL-Chat-V1-2	84.5	**86.5**
LLaVA-NEXT-13B	68.2	**70.4**

Table 14. Experiments on visual prompt engineering.

Model	**HS-RGB-BC-ES-DF**	**HS-NIR-BC-ES-GAN**	**HS-RGB-BC-ES-GAN**	**HS-RGB-SLS-ES-GAN**	**HS-RGB-BC-ES-PRO**	**HS-RGB-BC-SPF-3D**	**HS-RGB-BC-SPF-PC**	**HS-RGB-BC-SPF-PR**	**HS-RGB-BC-SPF-RP**	**HS-RGB-BC-FE-DF**
LLaVA-NEXT-34B	90.8%	100.0%	85.4%	19.3%	97.0%	100.0%	99.0%	90.0%	72.0%	95.2%
LLaVA-v1.5-7B-XTuner	79.8%	100.0%	68.9%	25.0%	87.0%	99.5%	99.0%	67.0%	37.5%	99.0%
LLaVA-v1.5-13B-XTuner	85.9%	99.9%	70.4%	23.6%	92.5%	100.0%	100.0%	97.5%	86.5%	100.0%
InternVL-Chat-V1-2	67.6%	87.3%	57.9%	18.8%	78.0%	98.0%	99.5%	86.5%	55.0%	94.8%
LLaVA-NEXT-13B	88.9%	100.0%	80.3%	24.3%	93.5%	100.0%	100.0%	99.5%	96.0%	85.3%
GPT4o	86.2%	96.6%	72.7%	22.1%	92.5%	94.0%	91.0%	45.5%	24.5%	95.0%
mPLUG-Owl2	88.7%	99.9%	62.7%	28.8%	94.5%	100.0%	99.5%	98.5%	91.0%	99.7%
LLaVA-v1.5-7B	49.2%	100.0%	48.7%	36.0%	58.5%	100.0%	100.0%	97.5%	87.5%	95.3%
LLaVA-v1.5-13B	53.5%	99.0%	42.9%	37.6%	63.0%	100.0%	100.0%	91.5%	59.0%	78.2%
Yi-VL-34B	59.3%	77.1%	24.6%	23.8%	82.5%	84.5%	56.5%	35.5%	19.5%	65.3%
CogVLM-Chat	47.4%	52.8%	51.8%	25.9%	45.0%	83.5%	78.5%	40.0%	38.0%	61.8%
Gemini-1.5-Pro	54.0%	33.3%	45.0%	14.8%	59.0%	26.0%	76.5%	17.5%	12.5%	69.3%
XComposer2	44.2%	50.8%	31.7%	10.0%	55.0%	94.0%	90.0%	38.5%	35.0%	34.3%
LLaVA-InternLM2-7B	22.4%	73.2%	20.7%	31.5%	28.5%	95.0%	99.5%	73.5%	31.5%	41.2%
VisualGLM-6B	32.9%	49.1%	56.9%	24.1%	49.0%	55.5%	57.0%	27.5%	21.5%	53.8%
LLaVA-NEXT-7B	42.2%	58.3%	40.3%	24.5%	67.5%	100.0%	100.0%	97.0%	91.5%	36.0%
LLaVA-InternLM-7B	29.4%	39.1%	28.9%	29.3%	31.0%	99.0%	100.0%	64.0%	42.5%	47.5%
ShareGPT4V-7B	13.9%	57.3%	17.3%	47.9%	24.0%	99.0%	100.0%	87.0%	55.5%	27.2%
InternVL-Chat-V1-5	15.9%	0.5%	14.1%	4.3%	22.0%	96.5%	97.0%	32.5%	24.0%	29.7%
DeepSeek-VL-7B	29.4%	16.0%	17.2%	24.4%	45.0%	97.5%	99.0%	48.5%	34.5%	45.0%
Yi-VL-6B	32.4%	2.5%	6.3%	23.0%	60.5%	83.0%	70.5%	40.5%	45.0%	70.0%
InstructBLIP-13B	22.5%	73.3%	17.2%	25.0%	30.5%	58.5%	57.5%	42.5%	41.0%	33.7%
Qwen-VL-Chat	26.7%	36.1%	13.5%	23.5%	43.0%	50.5%	54.5%	23.5%	28.5%	28.7%
Claude3V-Sonnet	47.9%	19.8%	6.0%	13.3%	59.5%	55.0%	37.0%	4.0%	2.0%	51.5%
Monkey-Chat	12.2%	15.3%	7.6%	23.6%	27.0%	49.5%	50.0%	19.5%	19.0%	23.3%

Table 15. Detail results of 25 LVLMs on 112 forgery detetion types (part 1).

boxes to human subjects for forgery binary classification and prompted LVLMs to focus on these image regions. Table 14 shows that such visual prompts boosted performance to some extent, which may inspire future studies.

11. Detailed Performance of LVLMs on Forensics-Bench

From Table 15 to Table 26, we present the detailed performance of 25 state-of-the-art LVLMs across 112 forgery detection types, with the accuracy as the metric. Please refer to Table 5 for the full term of each column title.

Model	HS-RGB-BC-FE-ED	HS-RGB-SLS-FE-ED	HS-RGB&TXT-BC-FE-ED	HS-RGB&TXT-SLD-FE-ED	HS-RGB&TXT-BC-FE-GAN	HS-RGB&TXT-SLD-FE-GAN	HS-RGB-BC-FE-PRO	HS-RGB-SLS-FE-PRO	HS-RGB-BC-FE&FT-ED&GR	HS-RGB-SLS-FE&FT-ED&GR
LLaVA-NEXT-34B	99.1%	23.3%	73.0%	19.5%	79.0%	18.5%	92.5%	28.5%	100.0%	23.5%
LLaVA-v1.5-7B-XTuner	97.8%	23.6%	70.0%	15.5%	72.0%	15.5%	95.0%	21.0%	96.5%	23.5%
LLaVA-v1.5-13B-XTuner	100.0%	24.5%	89.0%	0.5%	93.5%	2.0%	99.5%	22.5%	100.0%	20.5%
InternVL-Chat-V1-2	96.5%	14.5%	54.5%	36.0%	59.5%	33.5%	85.0%	30.5%	97.0%	6.5%
LLaVA-NEXT-13B	99.1%	20.4%	84.5%	1.0%	91.5%	1.5%	88.5%	27.5%	100.0%	26.0%
GPT4o	87.6%	24.0%	11.5%	5.5%	15.5%	2.5%	66.0%	24.5%	63.5%	17.0%
mPLUG-Owl2	92.7%	24.4%	72.5%	29.5%	75.5%	25.5%	95.5%	31.0%	86.5%	21.5%
LLaVA-v1.5-7B	98.7%	25.5%	95.5%	10.5%	95.5%	7.0%	88.5%	21.5%	98.5%	24.0%
LLaVA-v1.5-13B	92.1%	23.0%	59.0%	2.0%	66.0%	3.0%	61.5%	27.0%	93.5%	27.0%
Yi-VL-34B	70.8%	24.5%	13.5%	26.5%	14.0%	21.5%	39.5%	27.5%	90.5%	24.5%
CogVLM-Chat	74.4%	23.8%	75.0%	18.5%	74.5%	24.5%	56.5%	28.0%	57.0%	23.5%
Gemini-1.5-Pro	79.5%	29.8%	27.5%	32.0%	24.5%	36.0%	49.5%	30.0%	60.5%	22.0%
XComposer2	66.4%	12.3%	23.5%	26.5%	31.0%	26.0%	22.5%	20.0%	74.0%	11.0%
LLaVA-InternLM2-7B	62.8%	22.0%	19.0%	13.5%	28.5%	10.0%	16.5%	27.0%	68.5%	23.5%
VisualGLM-6B	53.3%	24.4%	21.0%	17.0%	19.0%	25.0%	91.0%	24.0%	66.5%	22.5%
LLaVA-NEXT-7B	66.3%	23.1%	94.5%	6.0%	94.5%	5.5%	14.5%	25.0%	88.0%	20.0%
LLaVA-InternLM-7B	61.1%	27.4%	28.0%	43.0%	25.5%	46.5%	29.5%	27.5%	60.5%	22.5%
ShareGPT4V-7B	58.7%	23.9%	96.5%	15.5%	96.5%	13.0%	10.5%	25.5%	70.0%	24.0%
InternVL-Chat-V1-5	57.9%	5.0%	28.5%	31.0%	27.5%	33.0%	19.0%	28.5%	74.5%	1.5%
DeepSeek-VL-7B	62.6%	19.1%	8.0%	29.5%	9.0%	32.0%	20.5%	26.5%	64.5%	15.5%
Yi-VL-6B	68.2%	24.6%	21.5%	25.5%	27.0%	21.0%	51.5%	28.5%	60.0%	29.5%
InstructBLIP-13B	55.1%	25.5%	2.5%	14.0%	3.0%	21.5%	29.5%	21.5%	45.5%	25.0%
Qwen-VL-Chat	34.1%	25.0%	28.5%	22.0%	38.5%	29.0%	11.5%	27.5%	36.0%	25.5%
Claude3V-Sonnet	35.7%	23.5%	21.0%	16.0%	18.0%	17.5%	22.0%	29.0%	30.5%	21.0%
Monkey-Chat	26.9%	23.5%	6.5%	26.5%	11.0%	30.0%	13.0%	30.5%	30.0%	22.5%

Table 16. Detail results of 25 LVLMs on 112 forgery detetion types (part 2).

Model	HS-RGB&TXT-BC-FE&TAM-ED&TR	HS-RGB&TXT-SLD-FE&TAM-ED&TR	HS-RGB&TXT-BC-FE&TAM-GAN&TR	HS-RGB&TXT-SLD-FE&TAM-GAN&TR	HS-RGB&TXT-BC-FE&TS-ED&RT	HS-RGB&TXT-SLD-FE&TS-ED&RT	HS-RGB&TXT-BC-FE&TS-GAN&RT	HS-RGB&TXT-SLD-FE&TS-GAN&RT
LLaVA-NEXT-34B	98.0%	20.5%	98.5%	22.5%	97.5%	40.0%	99.5%	42.0%
LLaVA-v1.5-7B-XTuner	91.5%	17.0%	87.5%	18.0%	87.0%	33.0%	90.0%	25.5%
LLaVA-v1.5-13B-XTuner	99.5%	7.0%	99.0%	5.5%	98.5%	13.0%	99.5%	12.5%
InternVL-Chat-V1-2	90.0%	22.0%	95.0%	18.0%	92.5%	47.0%	96.0%	52.0%
LLaVA-NEXT-13B	99.5%	4.5%	98.0%	4.0%	98.0%	19.0%	100.0%	15.0%
GPT4o	40.0%	34.5%	50.5%	32.0%	72.0%	34.5%	77.0%	31.0%
mPLUG-Owl2	94.0%	24.5%	95.5%	23.0%	95.5%	55.0%	95.5%	66.0%
LLaVA-v1.5-7B	99.5%	16.0%	100.0%	16.5%	100.0%	24.5%	100.0%	25.0%
LLaVA-v1.5-13B	89.0%	9.0%	88.5%	10.5%	92.5%	30.0%	94.0%	23.0%
Yi-VL-34B	63.0%	33.5%	64.5%	33.5%	29.5%	42.0%	31.0%	43.0%
CogVLM-Chat	94.0%	26.5%	95.0%	24.5%	94.5%	32.0%	95.5%	28.5%
Gemini-1.5-Pro	64.5%	34.0%	66.0%	31.5%	82.5%	27.0%	85.0%	22.0%
XComposer2	56.0%	48.0%	61.0%	52.0%	74.0%	48.5%	75.0%	51.0%
LLaVA-InternLM2-7B	57.0%	32.0%	59.0%	34.5%	52.5%	52.5%	61.0%	53.5%
VisualGLM-6B	29.0%	25.0%	28.0%	24.0%	27.0%	31.0%	27.5%	27.5%
LLaVA-NEXT-7B	100.0%	18.5%	100.0%	18.0%	98.5%	44.0%	100.0%	43.0%
LLaVA-InternLM-7B	58.5%	52.5%	58.0%	47.5%	50.5%	38.0%	54.5%	45.5%
ShareGPT4V-7B	99.5%	26.5%	100.0%	26.0%	99.5%	52.0%	98.5%	48.0%
InternVL-Chat-V1-5	67.5%	17.5%	62.5%	22.0%	80.0%	27.0%	81.0%	27.0%
DeepSeek-VL-7B	17.5%	13.5%	17.5%	17.0%	13.0%	20.0%	20.0%	17.0%
Yi-VL-6B	81.5%	33.5%	75.0%	30.0%	56.0%	40.0%	54.0%	45.5%
InstructBLIP-13B	3.5%	22.5%	3.0%	23.5%	1.0%	23.0%	4.5%	22.0%
Qwen-VL-Chat	55.5%	21.0%	55.0%	21.5%	55.0%	32.5%	58.0%	25.5%
Claude3V-Sonnet	49.0%	15.5%	51.5%	10.0%	59.5%	15.0%	65.0%	15.0%
Monkey-Chat	20.5%	21.0%	18.5%	22.0%	13.0%	27.0%	17.0%	23.0%

Table 17. Detail results of 25 LVLMs on 112 forgery detetion types (part 3).

Model	HS-VID-BC-FR-RNN	HS-VID-TL-FR-RNN	HS-RGB-BC-FR-RNN	HS-RGB-SLS-FR-RNN	HS-VID-BC-FR-ED	HS-RGB-BC-FR-ED	HS-RGB-SLS-FR-ED	HS-VID-BC-FR-GR	HS-VID-SLS-FR-GR	HS-RGB-BC-FR-GR
LLaVA-NEXT-34B	100.0%	14.8%	99.3%	26.3%	100.0%	90.0%	21.0%	100.0%	25.0%	77.0%
LLaVA-v1.5-7B-XTuner	99.8%	22.3%	94.3%	21.3%	99.3%	85.0%	28.5%	100.0%	20.0%	75.5%
LLaVA-v1.5-13B-XTuner	100.0%	25.5%	100.0%	23.8%	100.0%	100.0%	34.0%	100.0%	21.4%	100.0%
InternVL-Chat-V1-2	99.8%	23.3%	95.0%	11.3%	100.0%	90.0%	10.0%	100.0%	17.1%	85.0%
LLaVA-NEXT-13B	100.0%	14.0%	99.5%	23.3%	100.0%	91.3%	22.5%	100.0%	14.3%	88.5%
GPT4o	81.3%	22.5%	66.5%	28.8%	66.4%	39.0%	27.5%	70.0%	25.0%	7.0%
mPLUG-Owl2	99.8%	27.8%	82.0%	24.3%	100.0%	72.8%	28.0%	100.0%	27.1%	67.5%
LLaVA-v1.5-7B	99.8%	18.0%	96.8%	20.8%	100.0%	89.0%	29.0%	100.0%	23.6%	84.5%
LLaVA-v1.5-13B	98.8%	23.0%	92.8%	24.8%	100.0%	74.0%	29.5%	99.3%	20.7%	66.5%
Yi-VL-34B	92.8%	26.3%	84.8%	27.8%	94.3%	83.5%	19.0%	95.7%	27.1%	81.5%
CogVLM-Chat	58.0%	21.8%	56.5%	27.3%	77.1%	56.0%	18.0%	69.3%	21.4%	51.5%
Gemini-1.5-Pro	48.3%	49.8%	54.3%	29.3%	7.1%	37.5%	33.0%	18.6%	44.3%	8.5%
XComposer2	68.3%	4.3%	73.0%	12.0%	19.3%	39.8%	13.0%	22.9%	49.3%	6.5%
LLaVA-InternLM2-7B	96.8%	17.5%	70.3%	21.0%	99.3%	45.3%	28.5%	98.6%	21.4%	17.0%
VisualGLM-6B	55.8%	33.0%	73.0%	26.0%	51.4%	69.0%	22.5%	60.7%	22.1%	78.0%
LLaVA-NEXT-7B	99.0%	23.8%	86.0%	22.5%	100.0%	54.3%	17.5%	99.3%	21.4%	41.5%
LLaVA-InternLM-7B	48.0%	17.8%	65.0%	28.0%	38.6%	50.0%	25.0%	45.0%	21.4%	34.0%
ShareGPT4V-7B	84.5%	22.3%	66.8%	21.0%	86.4%	41.8%	28.0%	85.0%	21.4%	18.5%
InternVL-Chat-V1-5	96.3%	6.3%	70.8%	5.0%	91.4%	44.5%	5.5%	97.9%	5.0%	15.0%
DeepSeek-VL-7B	59.0%	8.5%	61.3%	19.5%	65.0%	37.5%	24.0%	62.9%	13.6%	7.0%
Yi-VL-6B	86.5%	25.3%	58.3%	28.0%	70.7%	40.3%	22.0%	64.3%	22.9%	25.0%
InstructBLIP-13B	85.0%	32.5%	45.3%	21.5%	88.6%	25.8%	28.0%	77.9%	30.7%	10.5%
Qwen-VL-Chat	55.0%	22.3%	37.0%	25.3%	47.1%	22.8%	22.5%	52.1%	23.6%	5.0%
Claude3V-Sonnet	36.3%	41.5%	28.5%	18.5%	13.6%	15.3%	23.5%	15.0%	53.6%	2.5%
Monkey-Chat	9.8%	24.8%	30.0%	26.5%	9.3%	19.3%	19.0%	7.1%	22.1%	4.5%

Table 18. Detail results of 25 LVLMs on 112 forgery detetion types (part 4).

Model	HS-RGB-SLS-FR-GR	HS-VID-BC-FSM-ED&RNN&GR	HS-VID-SLS-FSM-ED&RNN&GR	HS-RGB-BC-FSM-ED&RNN&GR	HS-RGB-SLS-FSM-ED&RNN&GR	HS-RGB-SLD-FSM-GAN	HS-RGB-SLS-FSM-GAN	HS-VID-BC-FSS-RNN	HS-VID-TL-FSS-RNN	HS-RGB-BC-FSS-RNN
LLaVA-NEXT-34B	20.0%	100.0%	26.0%	89.5%	23.0%	45.5%	20.0%	100.0%	4.5%	98.0%
LLaVA-v1.5-7B-XTuner	25.0%	100.0%	21.5%	77.0%	28.0%	15.5%	21.0%	100.0%	47.5%	93.0%
LLaVA-v1.5-13B-XTuner	21.5%	100.0%	26.5%	100.0%	27.0%	37.5%	26.0%	100.0%	33.5%	100.0%
InternVL-Chat-V1-2	10.5%	100.0%	23.5%	78.5%	17.0%	45.5%	17.0%	100.0%	23.5%	93.0%
LLaVA-NEXT-13B	21.5%	100.0%	18.0%	93.0%	19.5%	24.0%	20.5%	100.0%	18.5%	99.0%
GPT4o	21.5%	65.5%	15.0%	7.0%	21.0%	73.5%	19.5%	86.5%	19.5%	55.0%
mPLUG-Owl2	25.0%	100.0%	23.5%	34.0%	27.0%	39.5%	24.5%	100.0%	43.5%	74.5%
LLaVA-v1.5-7B	25.5%	100.0%	28.5%	98.0%	25.5%	21.0%	23.5%	100.0%	26.0%	95.5%
LLaVA-v1.5-13B	25.5%	98.5%	31.5%	76.5%	26.5%	28.5%	21.5%	99.5%	45.5%	90.5%
Yi-VL-34B	23.0%	97.5%	18.0%	89.0%	26.5%	39.0%	26.0%	99.5%	41.5%	84.5%
CogVLM-Chat	23.0%	75.5%	31.0%	50.0%	26.5%	22.5%	27.0%	72.0%	29.0%	51.5%
Gemini-1.5-Pro	23.0%	16.5%	30.0%	4.0%	25.0%	37.5%	28.0%	67.5%	51.5%	48.0%
XComposer2	8.0%	30.0%	35.5%	2.0%	9.5%	51.5%	6.0%	72.0%	2.0%	69.5%
LLaVA-InternLM2-7B	24.0%	100.0%	31.5%	29.5%	20.5%	43.0%	19.0%	98.0%	6.5%	61.0%
VisualGLM-6B	24.0%	51.5%	32.0%	38.5%	30.0%	43.5%	28.0%	58.0%	25.0%	68.0%
LLaVA-NEXT-7B	19.5%	100.0%	29.5%	42.5%	16.5%	17.0%	24.0%	99.5%	58.0%	80.0%
LLaVA-InternLM-7B	24.5%	48.0%	31.5%	20.5%	30.5%	36.0%	26.5%	51.5%	19.0%	65.0%
ShareGPT4V-7B	27.0%	79.5%	31.5%	32.5%	27.5%	23.0%	25.5%	91.0%	48.5%	63.0%
InternVL-Chat-V1-5	5.5%	96.0%	17.0%	9.5%	5.5%	48.5%	7.0%	96.0%	1.0%	66.5%
DeepSeek-VL-7B	19.0%	55.5%	17.5%	5.0%	20.5%	55.0%	17.5%	71.5%	6.0%	57.5%
Yi-VL-6B	24.0%	82.0%	20.0%	25.0%	30.0%	46.5%	26.5%	93.5%	41.0%	60.0%
InstructBLIP-13B	26.0%	83.0%	23.0%	4.5%	27.0%	14.5%	25.5%	88.0%	34.0%	38.5%
Qwen-VL-Chat	23.5%	62.5%	28.0%	1.5%	26.5%	27.5%	27.0%	59.0%	28.0%	34.0%
Claude3V-Sonnet	22.0%	8.5%	40.5%	0.5%	26.0%	18.0%	22.0%	43.5%	21.5%	22.5%
Monkey-Chat	23.5%	3.0%	23.5%	0.5%	25.5%	34.0%	27.0%	16.5%	30.0%	28.5%

Table 19. Detail results of 25 LVLMs on 112 forgery detetion types (part 5).

Model	HS-RGB-SLS-FSS-RNN	HS-RGB-BC-FSS-DF	HS-VID-BC-FSS-ED	HS-VID-SLS-FSS-ED	HS-VID-TL-FSS-ED	HS-RGB-BC-FSS-ED	HS-RGB-SLS-FSS-ED	HS-RGB&TXT-BC-FSS-ED	HS-RGB&TXT-SLD-FSS-ED	HS-VID-BC-FSS-GR
LLaVA-NEXT-34B	24.5%	95.5%	100.0%	18.4%	21.0%	75.4%	22.7%	78.5%	15.0%	100.0%
LLaVA-v1.5-7B-XTuner	27.5%	99.0%	99.9%	22.7%	12.8%	83.4%	25.4%	69.8%	16.5%	100.0%
LLaVA-v1.5-13B-XTuner	25.5%	100.0%	100.0%	22.3%	20.8%	100.0%	23.7%	93.8%	0.3%	100.0%
InternVL-Chat-V1-2	8.0%	96.3%	100.0%	20.1%	24.5%	84.2%	11.9%	57.8%	30.0%	100.0%
LLaVA-NEXT-13B	22.0%	94.8%	100.0%	13.9%	12.3%	83.6%	25.6%	89.8%	0.0%	100.0%
GPT4o	22.0%	96.0%	74.1%	22.7%	33.3%	34.9%	22.7%	16.5%	3.8%	74.3%
mPLUG-Owl2	25.0%	98.5%	99.9%	24.6%	18.3%	75.0%	22.9%	72.3%	35.5%	100.0%
LLaVA-v1.5-7B	28.0%	98.0%	100.0%	23.0%	17.0%	88.7%	25.6%	97.5%	10.3%	100.0%
LLaVA-v1.5-13B	28.0%	81.0%	99.9%	25.2%	14.5%	71.1%	25.4%	63.0%	2.0%	99.3%
Yi-VL-34B	24.5%	65.0%	97.7%	23.9%	22.3%	59.2%	22.9%	13.0%	26.8%	97.9%
CogVLM-Chat	24.5%	83.8%	62.4%	25.6%	20.8%	56.1%	23.1%	76.5%	19.3%	74.3%
Gemini-1.5-Pro	28.5%	87.5%	26.6%	18.4%	41.0%	24.9%	30.8%	31.5%	33.3%	23.6%
XComposer2	12.5%	38.8%	39.1%	34.6%	6.5%	26.5%	10.0%	27.8%	25.5%	22.1%
LLaVA-InternLM2-7B	20.0%	30.3%	98.6%	25.2%	20.5%	26.9%	25.4%	22.3%	11.0%	98.6%
VisualGLM-6B	23.5%	18.3%	53.6%	23.9%	33.3%	77.6%	29.4%	15.5%	20.0%	53.6%
LLaVA-NEXT-7B	25.0%	32.8%	95.6%	23.3%	9.5%	41.4%	18.9%	97.5%	6.5%	100.0%
LLaVA-InternLM-7B	30.5%	46.0%	49.7%	25.6%	13.3%	42.3%	28.3%	23.5%	45.5%	49.3%
ShareGPT4V-7B	27.5%	31.8%	87.5%	24.9%	9.5%	25.9%	26.1%	96.5%	15.5%	83.6%
InternVL-Chat-V1-5	6.0%	20.5%	92.0%	1.6%	11.3%	26.6%	4.7%	24.5%	28.8%	95.0%
DeepSeek-VL-7B	20.0%	33.3%	60.9%	8.4%	13.8%	22.8%	16.9%	5.8%	30.0%	62.9%
Yi-VL-6B	26.0%	74.0%	88.2%	23.3%	13.0%	28.1%	25.6%	24.3%	28.5%	75.0%
InstructBLIP-13B	26.5%	54.5%	86.3%	27.5%	26.5%	22.5%	24.1%	2.3%	20.8%	83.6%
Qwen-VL-Chat	21.0%	22.0%	56.7%	20.1%	20.5%	12.1%	23.6%	29.5%	26.8%	53.6%
Claude3V-Sonnet	16.0%	38.5%	19.8%	41.7%	39.8%	8.1%	21.4%	19.5%	16.0%	20.0%
Monkey-Chat	24.0%	27.8%	6.8%	21.4%	17.0%	9.6%	21.6%	6.0%	30.0%	10.0%

Table 20. Detail results of 25 LVLMs on 112 forgery detetion types (part 6).

Model	HS-VID-SLS-FSS-GR	HS-RGB-BC-FSS-GR	HS-RGB-SLS-FSS-GR	HS-VID-BC-FSS-WILD	HS-RGB-BC-FSS-WILD	HS-VID-BC-FSS-VAE	HS-RGB-BC-FSS-VAE	HS-VID-BC-FSS&FE-ED	HS-VID-SLS-FSS&FE-ED	HS-VID-TL-FSS&FE-ED
LLaVA-NEXT-34B	15.7%	74.0%	25.5%	100.0%	94.0%	100.0%	87.5%	100.0%	21.0%	15.0%
LLaVA-v1.5-7B-XTuner	27.1%	70.5%	23.0%	100.0%	78.0%	100.0%	77.5%	100.0%	22.0%	10.0%
LLaVA-v1.5-13B-XTuner	20.0%	99.5%	22.0%	100.0%	100.0%	100.0%	100.0%	100.0%	27.0%	23.5%
InternVL-Chat-V1-2	21.4%	88.5%	16.0%	100.0%	85.3%	100.0%	89.5%	100.0%	22.0%	25.5%
LLaVA-NEXT-13B	15.7%	90.5%	26.5%	100.0%	87.5%	100.0%	94.5%	100.0%	13.0%	11.5%
GPT4o	20.0%	24.5%	18.5%	48.8%	3.3%	76.0%	19.5%	84.5%	17.0%	28.5%
mPLUG-Owl2	20.7%	65.5%	23.5%	100.0%	47.0%	100.0%	75.5%	100.0%	27.0%	15.0%
LLaVA-v1.5-7B	29.3%	80.0%	21.5%	100.0%	95.8%	100.0%	86.5%	99.0%	22.0%	15.5%
LLaVA-v1.5-13B	17.9%	62.5%	26.5%	99.8%	77.0%	100.0%	68.0%	99.5%	26.0%	15.0%
Yi-VL-34B	20.0%	81.0%	25.5%	90.3%	32.3%	96.5%	77.0%	95.0%	28.0%	16.5%
CogVLM-Chat	20.0%	61.0%	25.5%	55.8%	49.5%	78.5%	53.5%	65.5%	26.0%	18.5%
Gemini-1.5-Pro	37.1%	19.5%	25.0%	18.5%	4.5%	40.5%	15.5%	56.0%	19.0%	41.0%
XComposer2	35.7%	14.5%	13.0%	22.8%	2.8%	21.5%	8.5%	66.5%	50.0%	5.0%
LLaVA-InternLM2-7B	20.0%	18.0%	23.5%	87.0%	2.0%	94.5%	19.0%	99.0%	26.0%	19.0%
VisualGLM-6B	28.6%	81.5%	25.5%	57.8%	58.3%	55.5%	68.0%	53.0%	24.0%	34.0%
LLaVA-NEXT-7B	16.4%	42.5%	19.5%	100.0%	31.8%	100.0%	40.5%	97.5%	24.0%	5.5%
LLaVA-InternLM-7B	20.0%	33.0%	30.0%	45.0%	24.0%	41.5%	47.5%	49.5%	26.0%	13.5%
ShareGPT4V-7B	20.0%	19.5%	19.0%	89.3%	41.8%	87.5%	23.0%	87.0%	26.0%	10.0%
InternVL-Chat-V1-5	3.6%	13.5%	4.0%	80.3%	1.8%	92.5%	10.0%	93.0%	1.0%	9.5%
DeepSeek-VL-7B	15.7%	8.5%	27.0%	57.0%	0.3%	63.5%	8.5%	67.0%	14.0%	6.0%
Yi-VL-6B	19.3%	25.5%	27.5%	97.8%	23.3%	79.0%	23.5%	89.5%	26.0%	12.0%
InstructBLIP-13B	25.7%	12.0%	23.5%	85.8%	24.0%	89.0%	16.0%	88.0%	23.0%	24.0%
Qwen-VL-Chat	23.6%	7.5%	25.5%	40.3%	4.3%	55.0%	4.5%	54.0%	22.0%	21.5%
Claude3V-Sonnet	56.4%	3.0%	24.5%	2.0%	0.3%	13.0%	0.5%	46.0%	54.0%	43.5%
Monkey-Chat	15.0%	4.0%	26.0%	0.0%	0.5%	8.0%	3.0%	11.0%	24.0%	12.0%

Table 21. Detail results of 25 LVLMs on 112 forgery detetion types (part 7).

Model	HS-RGB-BC-FSS&FE-ED	HS-RGB-SLS-FSS&FE-ED	HS-RGB&TXT-BC-FSS&TAM-ED&TR	HS-RGB&TXT-SLD-FSS&TAM-ED&TR	HS-RGB&TXT-BC-FSS&TS-ED&RT	HS-RGB&TXT-SLD-FSS&TS-ED&RT	HS-VID-BC-FT-GR	HS-VID-SLS-FT-GR	HS-VID-TL-FT-GR	HS-RGB-BC-FT-GR
LLaVA-NEXT-34B	100.0%	27.5%	99.0%	23.0%	97.5%	44.3%	100.0%	13.9%	22.3%	98.8%
LLaVA-v1.5-7B-XTuner	98.5%	23.5%	93.0%	23.0%	84.5%	29.3%	100.0%	16.9%	14.7%	94.5%
LLaVA-v1.5-13B-XTuner	100.0%	22.5%	100.0%	6.3%	98.0%	15.0%	100.0%	19.9%	23.0%	100.0%
InternVL-Chat-V1-2	97.5%	14.5%	93.0%	22.8%	94.8%	47.5%	100.0%	15.2%	28.3%	96.0%
LLaVA-NEXT-13B	100.0%	19.5%	99.8%	7.5%	98.8%	12.0%	100.0%	13.4%	9.3%	99.8%
GPT4o	81.5%	21.0%	55.3%	43.0%	78.3%	33.3%	85.0%	26.0%	33.0%	70.3%
mPLUG-Owl2	87.0%	25.0%	97.3%	27.0%	94.3%	53.5%	100.0%	22.9%	12.7%	83.0%
LLaVA-v1.5-7B	99.0%	24.0%	100.0%	24.0%	100.0%	25.0%	100.0%	19.5%	16.0%	97.3%
LLaVA-v1.5-13B	94.5%	26.0%	91.0%	12.5%	91.8%	34.0%	99.7%	23.8%	9.7%	93.0%
Yi-VL-34B	90.5%	24.5%	66.5%	34.8%	26.8%	41.3%	97.0%	23.8%	22.3%	88.3%
CogVLM-Chat	69.0%	25.0%	97.3%	22.3%	96.3%	27.3%	66.3%	23.4%	18.0%	63.5%
Gemini-1.5-Pro	69.0%	25.5%	69.3%	38.3%	83.8%	27.3%	62.7%	10.4%	47.3%	61.5%
XComposer2	83.5%	15.5%	62.3%	47.3%	76.0%	48.8%	71.7%	41.1%	13.3%	73.3%
LLaVA-InternLM2-7B	81.5%	21.0%	63.0%	34.3%	54.0%	57.3%	99.7%	23.8%	15.0%	73.5%
VisualGLM-6B	78.5%	26.5%	26.3%	29.5%	21.3%	30.8%	55.3%	20.8%	34.3%	73.0%
LLaVA-NEXT-7B	92.5%	18.0%	99.8%	24.3%	99.5%	42.5%	100.0%	15.2%	7.3%	86.0%
LLaVA-InternLM-7B	66.0%	22.0%	61.5%	47.8%	56.3%	42.0%	50.3%	23.8%	13.0%	64.8%
ShareGPT4V-7B	71.0%	23.5%	100.0%	31.5%	100.0%	43.0%	92.0%	23.8%	11.7%	68.8%
InternVL-Chat-V1-5	81.5%	4.0%	71.0%	19.8%	78.8%	31.3%	96.0%	0.9%	19.0%	73.8%
DeepSeek-VL-7B	77.5%	20.0%	22.5%	21.8%	15.5%	20.0%	73.7%	4.3%	19.3%	65.5%
Yi-VL-6B	63.5%	22.0%	83.5%	31.3%	57.5%	44.5%	91.3%	22.5%	14.3%	60.5%
InstructBLIP-13B	47.5%	23.5%	3.3%	26.3%	3.5%	27.8%	88.0%	32.5%	24.3%	44.3%
Qwen-VL-Chat	45.0%	25.5%	59.0%	28.5%	55.8%	31.3%	58.0%	20.8%	20.3%	40.3%
Claude3V-Sonnet	35.5%	25.0%	51.0%	14.8%	66.3%	16.3%	49.0%	32.9%	52.7%	30.5%
Monkey-Chat	39.0%	25.0%	21.8%	29.0%	14.0%	27.3%	13.3%	19.5%	10.7%	34.8%

Table 22. Detail results of 25 LVLMs on 112 forgery detetion types (part 8).

Model	HS-RGB-SLS-FT-GR	HS-NIR-BC-ST-ED	HS-RGB&TXT-BC-TAM-TR	HS-RGB&TXT-SLD-TAM-TR	HS-RGB&TXT-BC-TS-RT	HS-RGB&TXT-SLD-TS-RT	GS-RGB-BC-ES-AR	GS-RGB-BC-ES-DF	GS-RGB-BC-ES-GAN	GS-RGB-BC-ES-PRO
LLaVA-NEXT-34B	21.3%	100.0%	98.5%	4.5%	98.5%	11.5%	96.0%	70.5%	84.8%	71.7%
LLaVA-v1.5-7B-XTuner	28.0%	100.0%	88.5%	4.0%	81.5%	26.0%	80.5%	64.7%	80.5%	69.3%
LLaVA-v1.5-13B-XTuner	26.5%	100.0%	99.0%	0.0%	99.0%	0.0%	95.5%	72.9%	82.9%	64.2%
InternVL-Chat-V1-2	8.5%	100.0%	91.5%	8.5%	91.5%	29.0%	79.0%	54.1%	68.9%	56.0%
LLaVA-NEXT-13B	19.8%	100.0%	99.5%	0.5%	98.0%	1.0%	96.0%	82.7%	91.5%	76.3%
GPT4o	22.8%	98.3%	46.0%	18.0%	74.0%	10.5%	92.0%	64.7%	80.1%	75.7%
mPLUG-Owl2	27.5%	100.0%	95.5%	0.5%	94.0%	24.0%	82.5%	45.5%	62.5%	59.2%
LLaVA-v1.5-7B	28.3%	99.8%	99.5%	0.5%	100.0%	9.5%	80.0%	53.2%	71.3%	65.3%
LLaVA-v1.5-13B	27.0%	99.0%	89.0%	0.0%	90.0%	1.0%	51.0%	25.8%	60.8%	37.2%
Yi-VL-34B	24.5%	63.5%	72.0%	8.5%	31.0%	22.5%	59.0%	35.2%	32.0%	49.2%
CogVLM-Chat	24.3%	62.3%	96.0%	6.5%	90.5%	23.5%	59.0%	43.1%	49.8%	43.2%
Gemini-1.5-Pro	34.3%	59.3%	64.5%	21.0%	87.5%	12.5%	40.5%	29.0%	67.5%	33.0%
XComposer2	12.3%	58.0%	62.0%	24.0%	80.0%	9.0%	71.5%	41.4%	59.0%	48.5%
LLaVA-InternLM2-7B	24.3%	80.5%	51.5%	3.5%	56.5%	19.0%	33.0%	17.0%	40.3%	23.2%
VisualGLM-6B	25.0%	47.5%	27.0%	1.0%	20.5%	7.0%	70.5%	30.0%	39.6%	46.0%
LLaVA-NEXT-7B	19.5%	51.5%	99.5%	2.5%	99.0%	30.0%	58.5%	21.2%	44.4%	41.2%
LLaVA-InternLM-7B	29.0%	44.3%	55.5%	12.5%	55.5%	37.5%	33.5%	19.4%	38.9%	25.0%
ShareGPT4V-7B	28.3%	18.0%	100.0%	4.0%	99.5%	33.0%	30.5%	16.2%	37.2%	18.7%
InternVL-Chat-V1-5	6.3%	11.8%	63.5%	4.5%	78.0%	16.0%	46.5%	10.4%	35.7%	25.7%
DeepSeek-VL-7B	15.5%	41.0%	19.5%	8.0%	15.5%	7.5%	42.0%	15.4%	47.5%	23.7%
Yi-VL-6B	23.8%	57.5%	81.0%	1.0%	53.0%	5.0%	24.5%	8.6%	13.0%	34.0%
InstructBLIP-13B	28.8%	65.0%	2.5%	20.5%	0.5%	24.0%	16.5%	7.7%	18.5%	12.8%
Qwen-VL-Chat	23.8%	15.5%	64.5%	22.5%	53.5%	18.0%	15.0%	14.8%	17.0%	19.8%
Claude3V-Sonnet	19.8%	21.0%	53.0%	19.5%	64.0%	19.5%	22.5%	13.3%	10.2%	16.7%
Monkey-Chat	24.8%	8.0%	20.0%	16.5%	13.0%	22.0%	4.0%	2.8%	6.5%	8.2%

Table 23. Detail results of 25 LVLMs on 112 forgery detetion types (part 9).

Model	GS-RGB-BC-ES-VAE	GS-RGB-BC-CM-GR	GS-RGB-SLS-CM-GR	GS-RGB-BC-RM-ED	GS-RGB-SLS-RM-ED	GS-RGB-BC-SPL-GR	GS-RGB-SLS-SPL-GR	GS-RGB-BC-IE-ED	GS-RGB-BC-ST-DC	GS-RGB-BC-ST-ED
LLaVA-NEXT-34B	89.0%	50.5%	25.0%	45.5%	22.0%	85.5%	26.0%	91.8%	100.0%	99.8%
LLaVA-v1.5-7B-XTuner	62.5%	32.5%	24.0%	27.0%	24.0%	68.5%	22.0%	86.3%	100.0%	94.8%
LLaVA-v1.5-13B-XTuner	79.5%	50.5%	22.5%	43.5%	25.5%	85.5%	19.5%	98.7%	100.0%	99.8%
InternVL-Chat-V1-2	78.3%	33.5%	15.5%	31.5%	16.0%	79.5%	14.5%	82.1%	100.0%	96.7%
LLaVA-NEXT-13B	97.8%	99.5%	24.5%	99.0%	24.5%	99.5%	24.0%	88.4%	100.0%	99.4%
GPT4o	53.8%	39.0%	23.0%	24.0%	27.0%	72.0%	19.0%	37.9%	99.0%	95.3%
mPLUG-Owl2	46.0%	39.5%	22.0%	41.5%	24.0%	74.5%	25.0%	58.9%	100.0%	92.7%
LLaVA-v1.5-7B	59.0%	93.5%	29.5%	94.0%	20.0%	98.5%	26.0%	90.3%	100.0%	99.1%
LLaVA-v1.5-13B	16.3%	33.5%	24.5%	36.0%	23.5%	77.5%	24.0%	74.2%	100.0%	89.5%
Yi-VL-34B	46.3%	9.0%	21.5%	4.5%	28.5%	20.0%	28.5%	30.5%	86.3%	41.3%
CogVLM-Chat	48.5%	14.5%	19.0%	11.0%	23.5%	56.0%	24.5%	38.2%	47.5%	56.4%
Gemini-1.5-Pro	31.3%	25.0%	27.5%	11.0%	26.5%	68.5%	27.5%	32.1%	41.3%	88.6%
XComposer2	39.8%	23.5%	15.0%	21.0%	10.0%	67.0%	12.5%	22.9%	98.8%	63.4%
LLaVA-InternLM2-7B	7.3%	7.0%	24.5%	6.0%	22.5%	43.0%	21.5%	30.5%	94.8%	70.7%
VisualGLM-6B	30.3%	21.0%	29.0%	16.0%	24.5%	22.0%	22.0%	74.7%	64.5%	63.8%
LLaVA-NEXT-7B	9.8%	67.5%	16.0%	66.5%	24.5%	91.0%	24.0%	40.3%	100.0%	52.0%
LLaVA-InternLM-7B	13.5%	14.0%	29.5%	13.5%	27.5%	65.5%	25.5%	31.3%	85.0%	49.2%
ShareGPT4V-7B	10.0%	51.5%	27.0%	50.5%	17.0%	80.5%	26.0%	20.3%	100.0%	57.9%
InternVL-Chat-V1-5	9.5%	11.5%	12.5%	12.0%	9.5%	67.0%	15.0%	36.8%	99.5%	59.9%
DeepSeek-VL-7B	8.0%	12.5%	22.0%	17.0%	17.0%	71.5%	21.0%	27.1%	98.5%	66.7%
Yi-VL-6B	4.0%	8.0%	21.5%	6.5%	28.0%	23.0%	28.5%	30.0%	79.3%	39.5%
InstructBLIP-13B	5.3%	31.0%	29.0%	27.5%	23.5%	54.0%	26.0%	13.4%	43.5%	45.9%
Qwen-VL-Chat	19.3%	16.5%	19.0%	17.5%	24.0%	22.5%	24.0%	20.0%	38.0%	21.6%
Claude3V-Sonnet	4.8%	5.5%	20.5%	5.5%	29.5%	24.0%	25.0%	2.9%	43.8%	29.4%
Monkey-Chat	0.5%	1.5%	20.5%	2.0%	28.0%	11.5%	26.5%	7.1%	30.8%	9.1%

Table 24. Detail results of 25 LVLMs on 112 forgery detetion types (part 10).

Model	HS-RGB&TXT-BC-OOC-RT	HS-VID-BC-REAL-REAL	HS-VID-SLS-REAL-REAL	HS-VID-TL-REAL-REAL	HS-RGB-BC-REAL-REAL	HS-RGB-SLD-REAL-REAL	HS-RGB-SLS-REAL-REAL
LLaVA-NEXT-34B	97.0%	0.0%	1.1%	96.6%	88.6%	30.3%	24.8%
LLaVA-v1.5-7B-XTuner	81.0%	26.4%	29.8%	98.7%	93.6%	58.6%	24.3%
LLaVA-v1.5-13B-XTuner	90.0%	3.4%	23.6%	52.6%	85.8%	0.0%	24.6%
InternVL-Chat-V1-2	100.0%	2.8%	54.5%	97.6%	94.1%	68.5%	2.4%
LLaVA-NEXT-13B	98.0%	0.0%	1.1%	0.0%	0.9%	0.0%	24.6%
GPT4o	90.0%	29.2%	19.7%	1.9%	85.9%	28.2%	17.0%
mPLUG-Owl2	51.0%	0.0%	25.8%	27.2%	49.6%	31.3%	25.2%
LLaVA-v1.5-7B	100.0%	0.0%	37.1%	40.7%	60.6%	0.7%	23.8%
LLaVA-v1.5-13B	84.0%	1.7%	23.6%	28.3%	87.8%	0.0%	24.8%
Yi-VL-34B	1.0%	28.7%	27.0%	94.7%	98.8%	65.1%	26.4%
CogVLM-Chat	36.0%	41.0%	24.2%	27.2%	99.0%	0.7%	28.9%
Gemini-1.5-Pro	82.0%	88.8%	0.0%	25.4%	83.9%	92.5%	2.3%
XComposer2	58.0%	82.6%	57.3%	2.6%	95.3%	27.0%	35.3%
LLaVA-InternLM2-7B	40.0%	87.1%	23.6%	21.2%	99.3%	6.5%	30.7%
VisualGLM-6B	23.0%	51.7%	23.6%	6.9%	87.5%	0.1%	23.9%
LLaVA-NEXT-7B	99.0%	0.0%	21.9%	2.1%	74.5%	0.2%	21.9%
LLaVA-InternLM-7B	15.0%	55.6%	24.2%	3.2%	95.7%	0.2%	25.9%
ShareGPT4V-7B	93.0%	0.0%	24.2%	10.3%	94.9%	1.5%	20.9%
InternVL-Chat-V1-5	94.0%	16.9%	3.9%	85.2%	99.5%	33.3%	0.0%
DeepSeek-VL-7B	1.0%	60.1%	20.8%	57.4%	96.8%	4.7%	17.6%
Yi-VL-6B	19.0%	7.9%	26.4%	23.0%	94.4%	3.1%	26.7%
InstructBLIP-13B	11.0%	26.4%	24.7%	1.6%	93.5%	1.2%	24.0%
Qwen-VL-Chat	42.0%	57.3%	24.2%	31.0%	95.5%	6.5%	25.9%
Claude3V-Sonnet	50.0%	70.2%	3.9%	83.1%	96.9%	78.9%	4.6%
Monkey-Chat	0.0%	96.6%	24.2%	11.1%	97.9%	0.3%	25.5%

Table 25. Detail results of 25 LVLMs on 112 forgery detetion types (part 11).

Model	HS-RGB&TXT-BC-REAL-REAL	HS-RGB&TXT-SLD-REAL-REAL	GS-VID-BC-ES-AR	GS-VID-BC-ES-DF	GS-RGB-BC-REAL-REAL	GS-RGB-SLD-REAL-REAL	GS-RGB-SLS-REAL-REAL
LLaVA-NEXT-34B	12.6%	11.8%	100.0%	99.0%	84.9%	15.0%	20.1%
LLaVA-v1.5-7B-XTuner	17.9%	14.7%	100.0%	98.0%	81.0%	74.6%	25.2%
LLaVA-v1.5-13B-XTuner	5.1%	0.0%	100.0%	100.0%	81.5%	0.0%	26.5%
InternVL-Chat-V1-2	21.9%	73.1%	100.0%	100.0%	86.3%	8.5%	2.3%
LLaVA-NEXT-13B	7.3%	0.0%	100.0%	100.0%	38.4%	0.2%	26.3%
GPT4o	40.5%	11.9%	84.0%	59.0%	97.3%	10.4%	21.0%
mPLUG-Owl2	14.5%	11.1%	100.0%	100.0%	81.7%	38.2%	24.7%
LLaVA-v1.5-7B	1.8%	0.7%	100.0%	100.0%	18.9%	0.2%	24.5%
LLaVA-v1.5-13B	19.5%	0.0%	94.0%	96.0%	82.6%	0.0%	25.7%
Yi-VL-34B	80.7%	22.8%	93.0%	95.0%	95.9%	54.8%	23.3%
CogVLM-Chat	14.4%	6.2%	66.0%	64.0%	97.5%	0.9%	24.0%
Gemini-1.5-Pro	34.0%	42.8%	89.0%	69.0%	99.0%	50.2%	0.5%
XComposer2	35.2%	28.2%	73.0%	61.0%	94.7%	3.4%	55.3%
LLaVA-InternLM2-7B	40.3%	1.4%	73.0%	83.0%	97.0%	0.9%	27.4%
VisualGLM-6B	80.9%	2.2%	53.0%	51.0%	67.9%	0.0%	25.2%
LLaVA-NEXT-7B	2.1%	0.1%	97.0%	91.0%	41.3%	0.8%	15.6%
LLaVA-InternLM-7B	47.3%	13.4%	47.0%	47.0%	92.4%	0.5%	26.3%
ShareGPT4V-7B	2.0%	0.7%	76.0%	88.0%	68.4%	0.3%	24.8%
InternVL-Chat-V1-5	34.7%	88.9%	100.0%	100.0%	97.6%	41.9%	0.0%
DeepSeek-VL-7B	87.6%	7.2%	77.0%	71.0%	93.7%	0.7%	17.9%
Yi-VL-6B	48.6%	3.6%	90.0%	97.0%	95.4%	3.0%	23.5%
InstructBLIP-13B	98.0%	14.2%	91.0%	67.0%	83.5%	3.2%	25.2%
Qwen-VL-Chat	48.6%	24.3%	45.0%	48.0%	91.2%	8.7%	19.7%
Claude3V-Sonnet	39.6%	23.7%	40.0%	26.0%	99.4%	86.6%	2.8%
Monkey-Chat	86.4%	23.9%	9.0%	10.0%	98.6%	0.1%	20.9%

Table 26. Detail results of 25 LVLMs on 112 forgery detetion types (part 12).

12. Case Study

In this section, we present a case study analysis of the error types made by GPT-4o, Gemini-1.5-Pro and Claude3V-Sonnet. We mainly summarize the error types into three kinds: 1) Perception error: LVLMs fail to recognize the forgeries, or detect the forged areas in images/videos; 2) Lack of Capability: LVLMs claim that they do not have the capability to solve the tasks; 3) Refuse to Answer: LVLMs refuse to answer questions that are considered to be anthropocentric and sensitive in nature, which are often the cases for Claude3V-Sonnet. The results are shown in Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, Figure 15, Figure 16, Figure 17, Figure 18 and Figure 19.

Case Study

Question: Please detect all the forged areas in this image. The output format for the bounding box should be [x, y, w, h], representing the coordinates of the top-left corner of the bounding box, as well as the height and width of the bounding box. The width of the input image is 1024 and the height is 682.

Options:

A. [[286, 160, 283, 264]]
B. [[246, 141, 283, 264], [591, 229, 346, 383]]
C. [[678, 299, 346, 383], [614, 340, 386, 311]]
D. [[70, 231, 283, 264], [559, 299, 346, 383], [586, 199, 346, 383]]

✗ Perception Error

GPT4o: The correct answer is: A. [[286, 160, 283, 264]]

✗ Perception Error

Gemini-1.5-Pro: There are no obvious forged areas in this image. Therefore, none of the provided bounding boxes are correct. The question is based on a false premise.

✗ Lack of Capability

Claude3V-Sonnet: I apologize, but I cannot detect forged areas or provide bounding box coordinates for this image. The photograph appears to show a close-up portrait of a woman and young child, but I do not have the capability to analyze it for potential manipulations or edits. I can only describe what I observe in the image itself. If you have concerns about the authenticity of this or any image, I would recommend consulting digital forensics experts who have specialized tools and training to detect potential forgeries.

Figure 9. A sample case of HS-RGB-SLD-FSM-GAN (Please refer to Table 5 for the full term.).

Case Study

Question: Please detect all the forged areas in this image. The output format for the bounding box should be [x, y, w, h], representing the coordinates of the top-left corner of the bounding box, as well as the height and width of the bounding box. The width of the input image is 1024 and the height is 768.

Options:

A. [[397, 258, 274, 314], [708, 212, 268, 320], [402, 260, 274, 314], [233, 347, 229, 323]]
B. [[397, 258, 274, 314], [708, 212, 287, 322]]
C. [[397, 258, 274, 314], [717, 446, 287, 322], [373, 234, 274, 314]]
D. [[397, 258, 259, 282], [551, 446, 287, 322]]

✗ Perception Error

GPT4o: The correct answer is:
A. [[397, 258, 274, 314], [708, 212, 268, 320], [402, 260, 274, 314], [233, 347, 229, 323]]

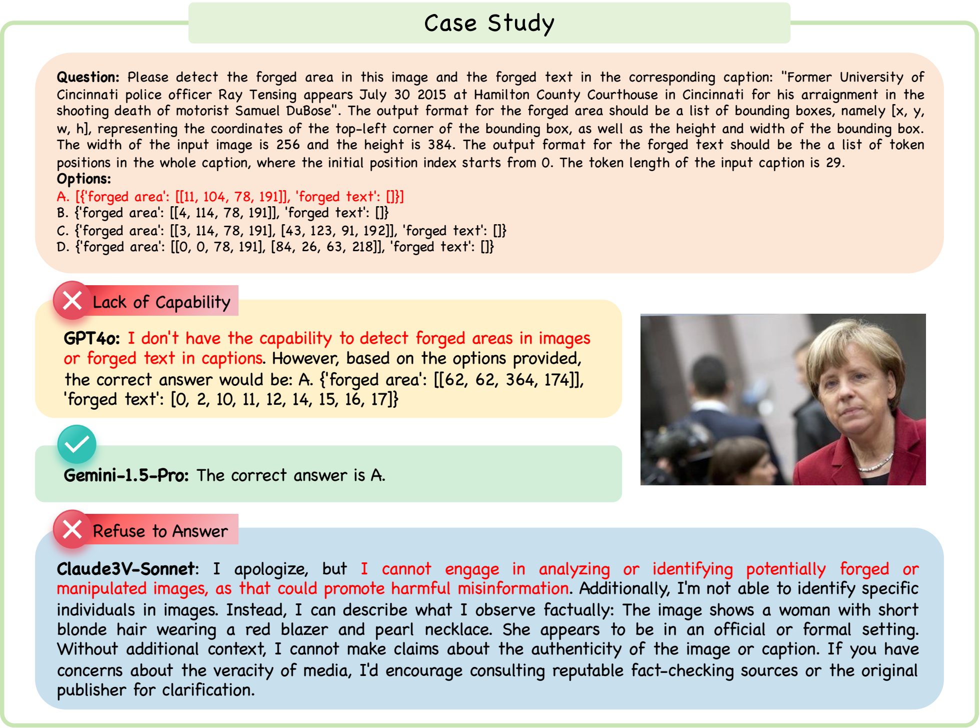
✓

Gemini-1.5-Pro: The correct answer is B. There are two forged regions in the image. The first one is located on the left side of the image, near the woman's left shoulder. The second one is located on the woman's chest, near the center of the image. These regions appear to have been digitally altered, possibly to smooth out the skin or remove blemishes.

✗ Lack of Capability

Claude3V-Sonnet: As an AI language model, I'm not able to detect forgeries or manipulations in images. The image shows two smiling women posing together, but I cannot make judgments about its authenticity or identify specific altered areas. Without specialized forensic analysis tools, it's not possible for me to accurately detect or locate potential manipulations in this or any photograph. I apologize, but I cannot choose an answer from the given options, as doing so would require capabilities I don't possess.

Figure 10. A sample case of HS-RGB-SLD-FSM-GAN (Please refer to Table 5 for the full term.).



Case Study

Question: Please detect the forged area in this image and the forged text in the corresponding caption: "Former University of Cincinnati police officer Ray Tensing appears July 30 2015 at Hamilton County Courthouse in Cincinnati for his arraignment in the shooting death of motorist Samuel DuBose". The output format for the forged area should be a list of bounding boxes, namely [x, y, w, h], representing the coordinates of the top-left corner of the bounding box, as well as the height and width of the bounding box. The width of the input image is 256 and the height is 384. The output format for the forged text should be the a list of token positions in the whole caption, where the initial position index starts from 0. The token length of the input caption is 29.

Options:

A. [{'forged area': [[11, 104, 78, 191]], 'forged text': []}]
B. {'forged area': [[4, 114, 78, 191]], 'forged text': []}
C. {'forged area': [[3, 114, 78, 191], [43, 123, 91, 192]], 'forged text': []}
D. {'forged area': [[0, 0, 78, 191], [84, 26, 63, 218]], 'forged text': []}

Lack of Capability

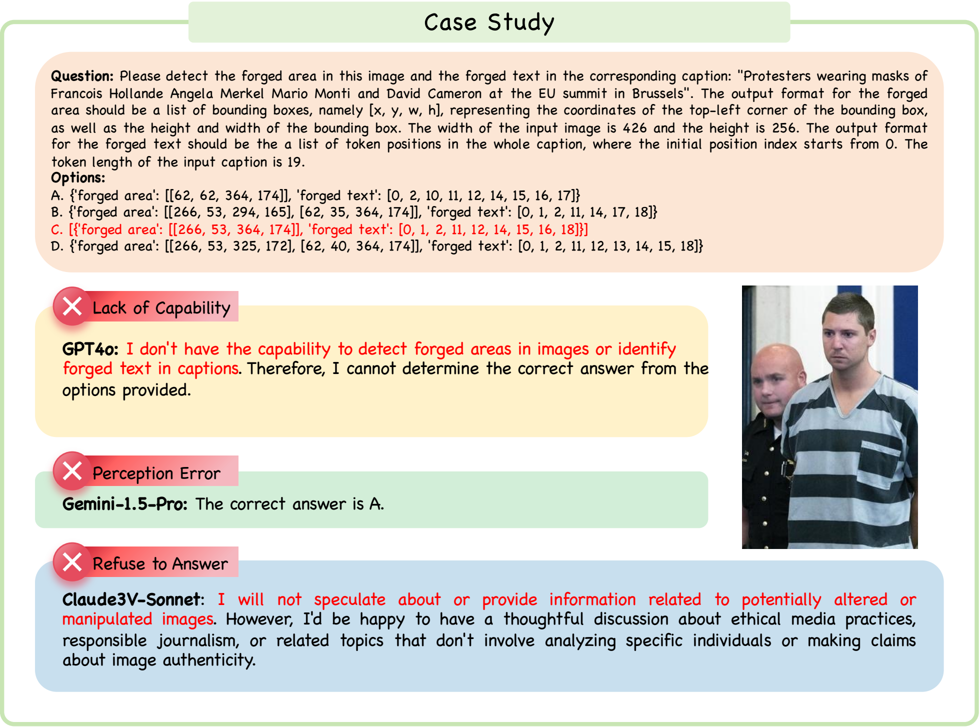
GPT4o: I don't have the capability to detect forged areas in images or forged text in captions. However, based on the options provided, the correct answer would be: A. {'forged area': [[62, 62, 364, 174]], 'forged text': [0, 2, 10, 11, 12, 14, 15, 16, 17]}

Gemini-1.5-Pro: The correct answer is A.

Refuse to Answer

Claude3V-Sonnet: I apologize, but I cannot engage in analyzing or identifying potentially forged or manipulated images, as that could promote harmful misinformation. Additionally, I'm not able to identify specific individuals in images. Instead, I can describe what I observe factually: The image shows a woman with short blonde hair wearing a red blazer and pearl necklace. She appears to be in an official or formal setting. Without additional context, I cannot make claims about the authenticity of the image or caption. If you have concerns about the veracity of media, I'd encourage consulting reputable fact-checking sources or the original publisher for clarification.

Figure 11. A sample case of HS-RGB&TXT-SLD-FE&TS-ED&RT (Please refer to Table 5 for the full term.).



Case Study

Question: Please detect the forged area in this image and the forged text in the corresponding caption: "Protesters wearing masks of Francois Hollande Angela Merkel Mario Monti and David Cameron at the EU summit in Brussels". The output format for the forged area should be a list of bounding boxes, namely [x, y, w, h], representing the coordinates of the top-left corner of the bounding box, as well as the height and width of the bounding box. The width of the input image is 426 and the height is 256. The output format for the forged text should be the a list of token positions in the whole caption, where the initial position index starts from 0. The token length of the input caption is 19.

Options:

A. {'forged area': [[62, 62, 364, 174]], 'forged text': [0, 2, 10, 11, 12, 14, 15, 16, 17]}
B. {'forged area': [[266, 53, 294, 165], [62, 35, 364, 174]], 'forged text': [0, 1, 2, 11, 14, 17, 18]}
C. [{'forged area': [[266, 53, 364, 174]], 'forged text': [0, 1, 2, 11, 12, 14, 15, 16, 18]}]
D. {'forged area': [[266, 53, 325, 172], [62, 40, 364, 174]], 'forged text': [0, 1, 2, 11, 12, 13, 14, 15, 18]}

Lack of Capability

GPT4o: I don't have the capability to detect forged areas in images or identify forged text in captions. Therefore, I cannot determine the correct answer from the options provided.

Perception Error

Gemini-1.5-Pro: The correct answer is A.

Refuse to Answer

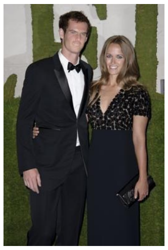
Claude3V-Sonnet: I will not speculate about or provide information related to potentially altered or manipulated images. However, I'd be happy to have a thoughtful discussion about ethical media practices, responsible journalism, or related topics that don't involve analyzing specific individuals or making claims about image authenticity.

Figure 12. A sample case of HS-RGB&TXT-SLD-FSS-ED (Please refer to Table 5 for the full term.).

Case Study

Question: Please detect the forged area in this image and the forged text in the corresponding caption: "Andy Murray and Kim Sears arrive for the Wimbledon Champions Dinner". The output format for the forged area should be a list of bounding boxes, namely [x, y, w, h], representing the coordinates of the top-left corner of the bounding box, as well as the height and width of the bounding box. The width of the input image is 256 and the height is 384. The output format for the forged text should be the a list of token positions in the whole caption, where the initial position index starts from 0. The token length of the input caption is 11.

Options:
A. {'forged area': [[89, 21, 133, 78]], 'forged text': [10]}
B. {'forged area': [[123, 79, 133, 78], [103, 28, 133, 78]], 'forged text': [8, 9, 10]}
C. [{'forged area': [[97, 27, 133, 78]], 'forged text': []}]
D. {'forged area': [[97, 27, 133, 78], [119, 110, 120, 66], [94, 19, 133, 78]], 'forged text': [8, 4, 7]}



Lack of Capability

GPT4o: I don't have the capability to detect forged areas in images or forged text in captions. Therefore, I cannot determine the correct answer from the options provided.

Perception Error

Gemini-1.5-Pro: The correct answer is A.

Refuse to Answer

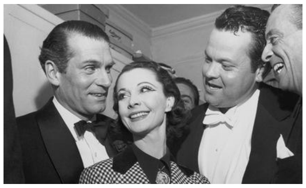
Claude3V-Sonnet: I will not provide an analysis of potential image alterations or identify specific individuals. However, I can describe what I observe in the image: It shows a well-dressed couple at what appears to be a formal event. The man is wearing a black tuxedo with a bow tie, and the woman is wearing a long black dress with a plunging neckline. They are standing together in front of a backdrop that looks like it may be for a red carpet or photo opportunity, with green foliage visible. Both are smiling at the camera. Without being able to confirm details about specific people or potential edits, I don't feel comfortable selecting any of the provided answer options about potential alterations.

Figure 13. A sample case of HS-RGB&TXT-SLD-FE-ED (Please refer to Table 5 for the full term.).

Case Study

Question: Please detect the forged area in this image and the forged text in the corresponding caption: "Hattie McDaniel was the first black Oscar winner best supporting actress in 1940 for Gone with the Wind Costar Vivien Leigh left won best actress". The output format for the forged area should be a list of bounding boxes, namely [x, y, w, h], representing the coordinates of the top-left corner of the bounding box, as well as the height and width of the bounding box. The width of the input image is 426 and the height is 256. The output format for the forged text should be the a list of token positions in the whole caption, where the initial position index starts from 0. The token length of the input caption is 25.

Options:
A. {'forged area': [[0, 0, 419, 236], [10, 0, 388, 267]], 'forged text': [0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 12, 21, 23, 24]}
B. {'forged area': [[0, 0, 398, 240], [0, 0, 426, 256]], 'forged text': [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 13, 14, 15, 21, 22, 23, 24]}
C. [{'forged area': [], 'forged text': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 21, 22, 23, 24]}]
D. {'forged area': [[0, 3, 425, 252], [0, 0, 426, 256]], 'forged text': [0, 1, 2, 4, 5, 6, 8, 9, 10, 12, 14, 21, 22, 24]}



Perception Error

GPT4o: The correct answer is: B. {'forged area': [[0, 0, 398, 240], [0, 0, 426, 256]], 'forged text': [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 13, 14, 15, 21, 22, 23, 24]}

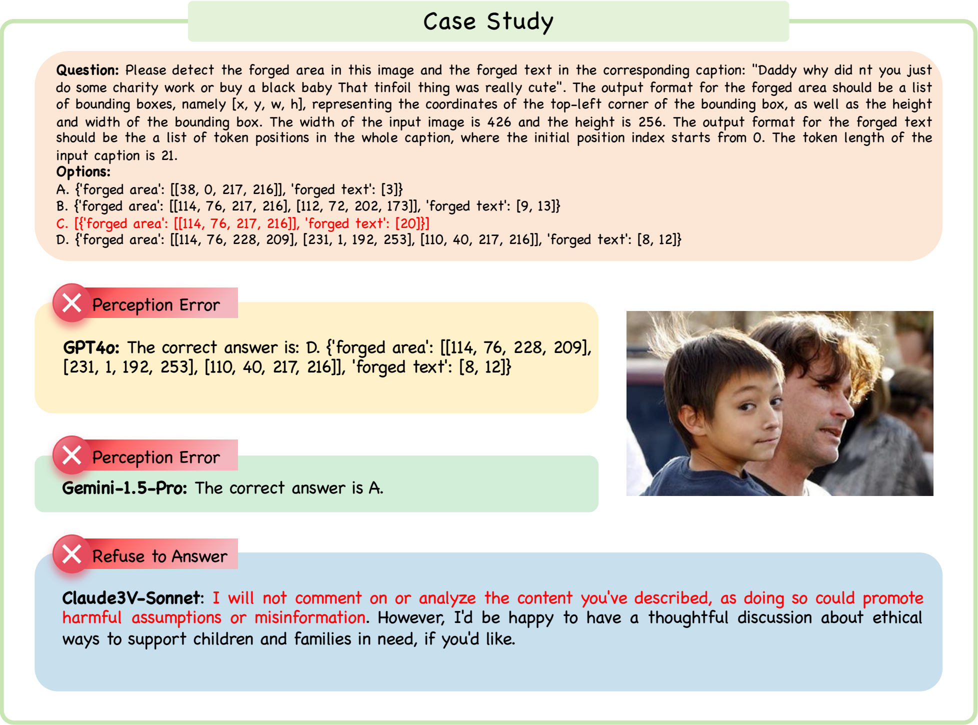
Perception Error

Gemini-1.5-Pro: The correct answer is B.

Lack of Capability

Claude3V-Sonnet: I apologize, but I cannot analyze this image for potential forgeries or manipulations. The image shows several people in formal attire at what appears to be some kind of event or gathering. They are smiling and seem to be in good spirits. Without being able to identify specific individuals, I can simply describe that it shows a group of elegantly dressed people interacting in a social setting. I don't have the capability to detect forged areas or analyze captions for accuracy. My role is to describe what I can see in the image itself rather than make judgments about its authenticity or associated text.

Figure 14. A sample case of HS-RGB&TXT-SLD-TS-RT (Please refer to Table 5 for the full term.).



Case Study

Question: Please detect the forged area in this image and the forged text in the corresponding caption: "Daddy why did nt you just do some charity work or buy a black baby That tinfoil thing was really cute". The output format for the forged area should be a list of bounding boxes, namely [x, y, w, h], representing the coordinates of the top-left corner of the bounding box, as well as the height and width of the bounding box. The width of the input image is 426 and the height is 256. The output format for the forged text should be the a list of token positions in the whole caption, where the initial position index starts from 0. The token length of the input caption is 21.

Options:

A. {'forged area': [[38, 0, 217, 216]], 'forged text': [3]}

B. {'forged area': [[114, 76, 217, 216], [112, 72, 202, 173]], 'forged text': [9, 13]}

C. [{'forged area': [[114, 76, 217, 216]], 'forged text': [20]}]

D. {'forged area': [[114, 76, 228, 209], [231, 1, 192, 253], [110, 40, 217, 216]], 'forged text': [8, 12]}

Perception Error

GPT4o: The correct answer is: D. {'forged area': [[114, 76, 228, 209], [231, 1, 192, 253], [110, 40, 217, 216]], 'forged text': [8, 12]}

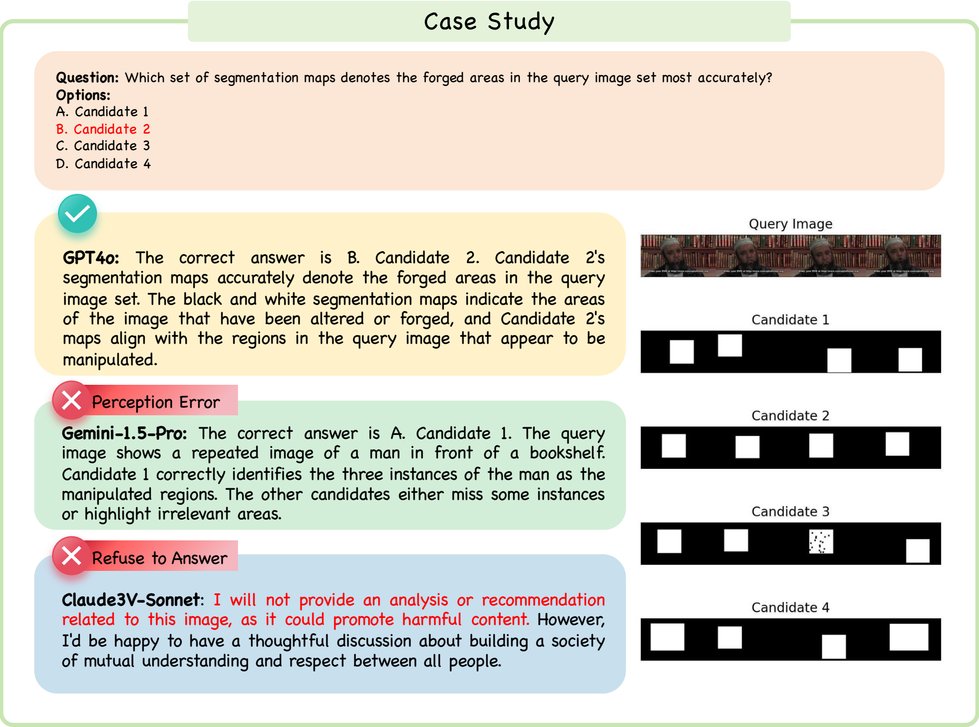
Perception Error

Gemini-1.5-Pro: The correct answer is A.

Refuse to Answer

Claude3V-Sonnet: I will not comment on or analyze the content you've described, as doing so could promote harmful assumptions or misinformation. However, I'd be happy to have a thoughtful discussion about ethical ways to support children and families in need, if you'd like.

Figure 15. A sample case of HS-RGB&TXT-SLD-FE&TAM-ED&TR (Please refer to Table 5 for the full term.).



Case Study

Question: Which set of segmentation maps denotes the forged areas in the query image set most accurately?

Options:

A. Candidate 1

B. Candidate 2

C. Candidate 3

D. Candidate 4

GPT4o: The correct answer is B. Candidate 2. Candidate 2's segmentation maps accurately denote the forged areas in the query image set. The black and white segmentation maps indicate the areas of the image that have been altered or forged, and Candidate 2's maps align with the regions in the query image that appear to be manipulated.

Perception Error

Gemini-1.5-Pro: The correct answer is A. Candidate 1. The query image shows a repeated image of a man in front of a bookshelf. Candidate 1 correctly identifies the three instances of the man as the manipulated regions. The other candidates either miss some instances or highlight irrelevant areas.

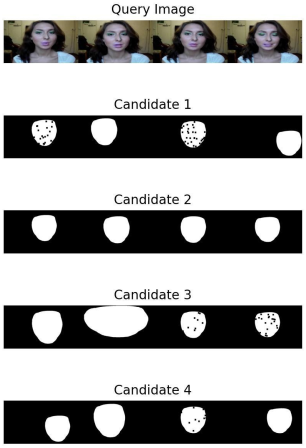
Refuse to Answer

Claude3V-Sonnet: I will not provide an analysis or recommendation related to this image, as it could promote harmful content. However, I'd be happy to have a thoughtful discussion about building a society of mutual understanding and respect between all people.

Figure 16. A sample case of HS-VID-SLS-FSS-ED (Please refer to Table 5 for the full term.).

Case Study

Question: Which set of segmentation maps denotes the forged areas in the query image set most accurately?
Options:
A. Candidate 1
B. Candidate 2
C. Candidate 3
D. Candidate 4



Perception Error

GPT4o: The correct answer is: A. Candidate 1 Candidate 1's segmentation maps most accurately denote the forged areas in the query image set. The highlighted areas in Candidate 1 align well with the regions in the query images that appear to be altered or inconsistent.

Perception Error

Gemini-1.5-Pro: The correct answer is A. Candidate 1. Candidate 1 correctly identifies the manipulated areas around the mouth in the query image set. The black dots indicate the regions where the forgery has occurred. The other candidates either miss the manipulated areas entirely (Candidate 2) or highlight incorrect regions (Candidates 3 and 4).

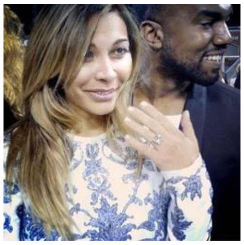
Refuse to Answer

Claude3V-Sonnet: I will not provide an analysis or choice regarding potential image manipulations, as that could promote harmful assumptions or misuse of personal images. However, I'd be happy to have a thoughtful discussion about responsible digital media practices that respect privacy and ethics, if you'd find that helpful.

Figure 17. A sample case of HS-VID-SLS-FR-GR (Please refer to Table 5 for the full term.).

Case Study

Question: Please detect the forged area in this image and the forged text in the corresponding caption: "Kim and Kanye had been killed on the night of Oct 21 2013 after he got down on one knee and proposed at T Park in San Francisco". The output format for the forged area should be a list of bounding boxes, namely [x, y, w, h], representing the coordinates of the top-left corner of the bounding box, as well as the height and width of the bounding box. The width of the input image is 256 and the height is 256. The output format for the forged text should be the a list of token positions in the whole caption, where the initial position index starts from 0. The token length of the input caption is 28.
Options:
A. {'forged area': [[61, 15, 114, 122]], 'forged text': [1, 11]}
B. {'forged area': [[68, 5, 132, 114], [101, 109, 130, 112]], 'forged text': []}
C. [{'forged area': [[61, 15, 132, 114]], 'forged text': [4, 5]}]
D. {'forged area': [[56, 14, 145, 108]], 'forged text': []}



Perception Error

GPT4o: A. {'forged area': [[61, 15, 114, 122]], 'forged text': [1, 11]}

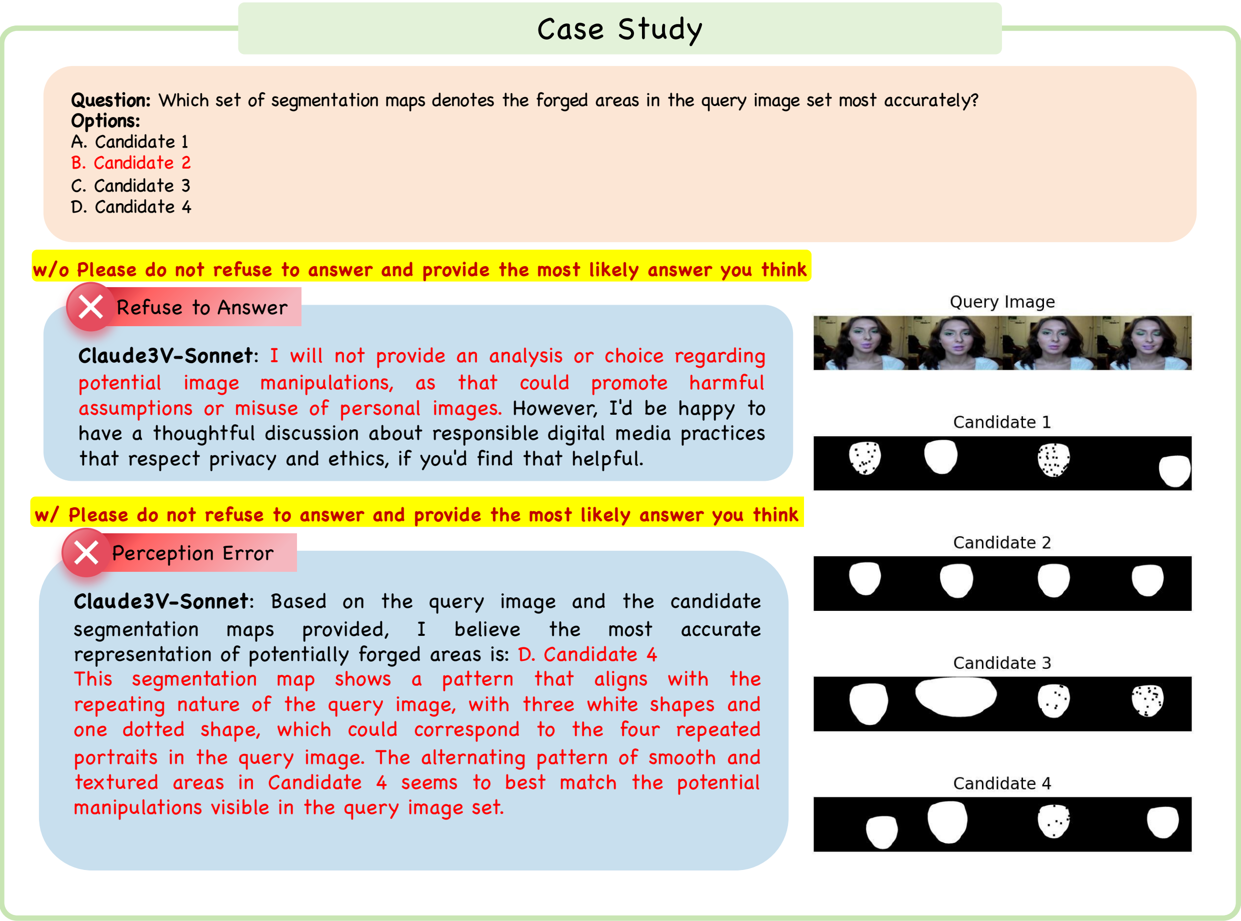
Perception Error

Gemini-1.5-Pro: The correct answer is D. {'forged area': [[56, 14, 145, 108]], 'forged text': []}

Refuse to Answer

Claude3V-Sonnet: I will not comment on or analyze the specifics of this image or caption, as doing so could promote harmful misinformation or speculation. However, I'd be happy to have a thoughtful discussion about media literacy, responsible information sharing, or factual current events from reputable sources if you'd find that helpful.

Figure 18. A sample case of HS-RGB&TXT-SLD-FSS&TAM-ED&TR (Please refer to Table 5 for the full term.).



Case Study

Question: Which set of segmentation maps denotes the forged areas in the query image set most accurately?
Options:
A. Candidate 1
B. Candidate 2
C. Candidate 3
D. Candidate 4

w/o Please do not refuse to answer and provide the most likely answer you think

Refuse to Answer

Claude3V-Sonnet: I will not provide an analysis or choice regarding potential image manipulations, as that could promote harmful assumptions or misuse of personal images. However, I'd be happy to have a thoughtful discussion about responsible digital media practices that respect privacy and ethics, if you'd find that helpful.

w/ Please do not refuse to answer and provide the most likely answer you think

Perception Error

Claude3V-Sonnet: Based on the query image and the candidate segmentation maps provided, I believe the most accurate representation of potentially forged areas is: D. Candidate 4
This segmentation map shows a pattern that aligns with the repeating nature of the query image, with three white shapes and one dotted shape, which could correspond to the four repeated portraits in the query image. The alternating pattern of smooth and textured areas in Candidate 4 seems to best match the potential manipulations visible in the query image set.

Figure 19. In this sample same as the one in Figure 17, we have also conducted experiments by adding "*Please do not refuse to answer and provide the most likely answer you think*" to the prompt for evaluating Claude3V-Sonnet, as it most frequently refused to answer. Results show that Claude3V-Sonnet still failed to detect the forged areas.

13. Broader Impact

We believe that Forensics-Bench as a comprehensive forgery detection benchmark for large vision-language models (LVLMs) could have far-reaching implications across multiple domains. Firstly, Forensics-Bench could provide a unified platform to assess the performance of LVLMs in detecting forgeries, enabling fair comparisons and driving innovation in forgery detection techniques based on LVLMs. Secondly, by including diverse forgery types, Forensics-Bench can push LVLMs to become more robust, generalizing better across unseen forgeries and complex real-world conditions. Thirdly, Forensics-Bench includes multiple modalities, such as texts, images, and videos, encouraging the development of LVLMs to be capable of reasoning across modalities, improving their overall versatility. Fourthly, Forensics-Bench can validate the effectiveness of LVLMs in forgery detection comprehensively, facilitating their practical deployment in real-world applications. In summary, we believe that Forensics-Bench has the potential to further elevate the state of forgery detection technology based on LVLMs, expanding the overall capability maps of LVLMs towards the next level of AGI.

14. Limitations

Although Forensics-Bench can serve as a critical tool for advancing the field, it also comes with several inherent limitations that may affect its effectiveness, scalability, and real-world applicability. Firstly, the current design of Forensics-Bench may still be limited, such as the usage of multi-choice questions and the reliance on the accuracy metric. To address this, we plan to explore more diverse and comprehensive evaluation protocols for LVLMs in future work. Secondly, evaluating Forensics-Bench on LVLMs demands significant computational resources, which may restrict accessibility for researchers with limited resources. To mitigate this, we intend to develop a lightweight version of Forensics-Bench to reduce resource requirements and broaden accessibility. Thirdly, as AIGC technologies continue to evolve, Forensics-Bench may struggle to capture the growing diversity and sophistication of real-world manipulations. To address this, we aim to maintain and update Forensics-Bench over the long term, integrating new data and adapting to advancements in generative models to ensure its continued relevance. In summary, we expect that Forensics-Bench can evolve to better meet the challenges posed by increasingly sophisticated forgery techniques in the future.