Mask-Guided, Training-Free Spatial Control for InstructPix2Pix

Detailed presentation notes (10-minute talk)

1) Problem & Motivation (use analogies)

- Pain point: Instruction-following editors (InstructPix2Pix) can repaint the whole scene. Asking "make the jacket red" can also stain the background.
- Analogy: Spray-painting without painter's tape—you color the table too. We need painter's tape for diffusion: a mask.
- Constraint: No time/compute to retrain a mask-aware diffusion model. Need a training-free fix.

2) Base Papers (what we stand on)

- InstructPix2Pix (Brooks et al., CVPR'23): learns text-conditioned edits via synthetic (before/after + instruction) pairs. Follows language well; lacks "where" control.
- Segment Anything (SAM, Kirillov et al., ICCV'23): promptable segmentation (click/box) for crisp masks on almost any object. Analogy: universal scissors that cut whatever you point at.

3) Our Idea: Mask-Blended Inference (training-free)

- Key move: do not touch weights. Run InstructPix2Pix normally, then blend its output with the original using the SAM mask.
- Equation: $\tilde{y} = M \odot y_{\text{baseline}} + (1 - M) \odot x_0$, where M is the mask (1 inside, 0 outside).
- Analogy: Painter's tape + airbrush; tape protects background, only exposed area gets painted.
- Why post-hoc blending works here: cheap (no training), deterministic (fixed seeds), fast (reuse public checkpoints).

4) Implementation Highlights (code map)

- Entrypoint: `submission/code/project_run.py` — loads SAM (`models/sam_vit_h_4b8939.pth`), InstructPix2Pix (`timbrooks/instruct-pix2pix`); resizes to 512×512; 20 steps; seeds fixed per case; saves original, mask, baseline, blended.
- Mask handling: nearest-neighbor resize; normalized to [0,1] to avoid overflow.
- Metrics: `submission/code/evaluate_metrics.py` — mIoU (spatial precision via change map vs. mask), CLIP (instruction fidelity).
- Runner scripts: `download_models.sh` (fetch SAM), `run_pipeline.sh` (deps + inference + metrics), `clean_run.sh` (optional reset).
- Data: `data/images/` — shirted woman, dog on grass, car on street.

5) Experiments (Colab A100 run)

- Setup: Colab, A100 40 GB, ∼2 min total after downloads.
- Cases (20 steps, guidance 1.5): Shirt → red leather; Dog → playful robot; Car → glowing hovercraft.
- Quantitative (mIoU, CLIP):
  - Shirt: 0.183, 0.206
  - Dog: 0.265, 0.261
  - Car: 0.173, 0.221
- Interpretation: Dog scores highest (clear silhouette); car shows spatial containment but a more ambitious semantic change.
- Qualitative: show rows (original — mask — baseline — blended). Note background preservation vs. baseline bleed.

6) Strengths, Limitations, Next Steps

- Strengths: training-free; uses public checkpoints; deterministic seeds; simple blend enforces spatial control.
- Limitations: post-hoc (does not steer the denoising trajectory); depends on SAM quality; heavy checkpoints.
- Next steps: attention/latent masking to guide trajectory; mask dilations/soft edges; more cases; optional human study on spatial fidelity.

7) Reproducibility & Submission

- Commands: `bash download_models.sh`; `bash run_pipeline.sh`.
- Assets: SAM checkpoint in `models/sam_vit_h_4b8939.pth`; sample images already in repo; results in `results/` and `submission/report/figures/`.
- Report: CVPR format (6–9 pages); code separate; GitHub link: https://github.com/abrahamyirga/Vision-Project

8) Q&A Cheat Sheet

- Why not train a mask-aware model? Time/compute; training-free gets 80/20 benefit fast.
- Does blending lose semantics? Inside mask we keep edited pixels; CLIP scores show instruction alignment; baseline preserved outside.
- How robust is SAM? Good on clear objects; fails if click misses or scene is cluttered—could add multi-click or dilation.
- Can this run on CPU? Yes, but slow; recommend GPU/Colab.

- Future work? Attention/latent masking, softer masks, more cases, human study.