

Supplementary Materials: Streamable Portrait Video Editing with Probabilistic Pixel Correspondence.

Anonymous Authors

1 PROVIDES MORE IMPLEMENTATION DETAILS OF PROBABILISTIC PIXEL CORRESPONDENCE ESTIMATION (PPCE)

1.1 Module Details

The proposed PPCE module comprises three main components: feature extraction, landmarks attention, and two output heads. The feature extraction component leverages the DINOv2 [4] method to extract texture features from input frames, while the mediapipe [2] framework is employed to extract 486 facial keypoints, providing precise information about facial landmarks such as the position of the eyes, nose, and mouth.

The landmarks interpolation component plays a crucial role in aligning the facial keypoints between the reference image and the current frame. By employing dense interpolation using triangular center of gravity interpolation with sparse face keypoints, the module achieves accurate transformation and warping of the reference frame features to match the corresponding regions in the current frame.

To establish the correspondence between the current frame and the reference frame, the module utilizes cross attention. Specifically, the features extracted from the current frame serve as query, while the features from the reference frame and the transformed reference frame features (transformed through landmarks interpolation) act as keys and values simultaneously. This enables the module to learn the matching relationship between the query features and the reference features.

The module includes two output heads, each serving a distinct purpose. The first head predicts the displacement map, which provides precise information about the movement or shifting between the reference frame and the current frame. The second head predicts the uncertainty map, which indicates the model's confidence level in its predictions. The architecture employed for these output heads is DPT decoder [7], which is well-suited for generating dense predictions.

1.2 Triangular Barycentric Interpolation of Landmarks

To establish the transformation between the current image (I_{curr}) and the reference image (I_{ref}), we can directly compute the transformation for landmarks by calculating the differences (delta) in their x and y coordinates.

For the remaining facial points, we can utilize triangular barycentric interpolation based on the neighboring landmarks to estimate their corresponding transformations. Let's denote the delta in x and y coordinates for a given landmark $P_{landmark}$ in the reference image and its corresponding landmark $P'_{landmark}$ in the current image as $\Delta x_{landmark}$ and $\Delta y_{landmark}$.

To compute the delta in x and y coordinates for any point within a triangle defined by three landmarks, P_{ref}^i and P_{ref}^j and P_{ref}^k in the

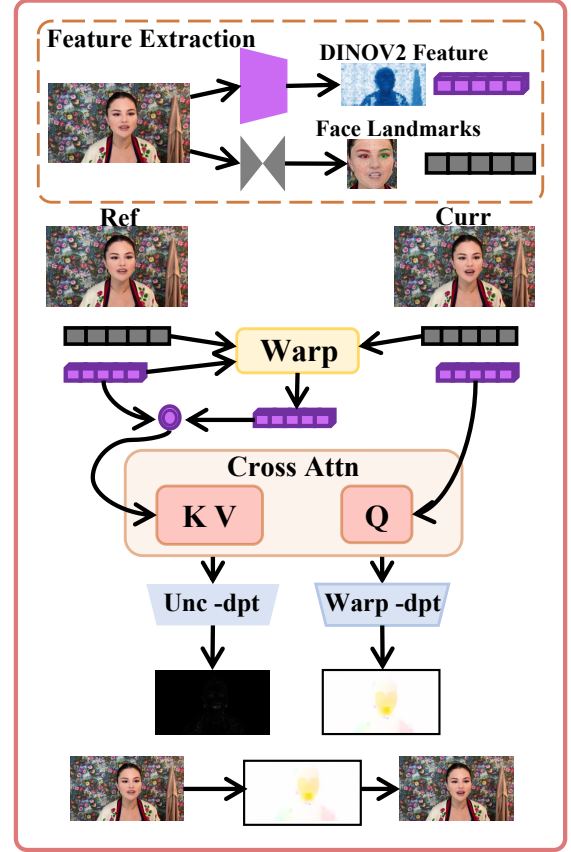


Figure 1: Provides more implementation details of Probabilistic Pixel Correspondence Estimation (PPCE). PPCE module comprises three main components: feature extraction, landmarks attention, and two output heads.

reference image, we use the barycentric coordinates (α, β, γ) as well as the deltas for the corresponding landmarks in the current image.

The formula to compute the delta in x and y coordinates for a pixel coordinate (u, v) within the triangle is as follows:

$$\Delta x = \alpha \cdot \Delta x_{ref}^i + \beta \cdot \Delta x_{ref}^j + \gamma \cdot \Delta x_{ref}^k \quad (1)$$

$$\Delta y = \alpha \cdot \Delta y_{ref}^i + \beta \cdot \Delta y_{ref}^j + \gamma \cdot \Delta y_{ref}^k \quad (2)$$

Here, Δx_{ref}^i , Δx_{ref}^j and Δx_{ref}^k represent the delta in x coordinates for the landmarks P_{ref}^i , P_{ref}^j and P_{ref}^k , respectively. Similarly, Δy_{ref}^i , Δy_{ref}^j and Δy_{ref}^k represent the delta in y coordinates for the corresponding landmarks.

By applying this formula to each pixel coordinate within the triangles defined by the landmarks, we can estimate the delta in x and y coordinates for the remaining facial points, thus achieving dense correspondence between the current and reference images.

Table 1: Comparative Preference Results on Diverse Web Videos.

Model	Motion Consistency \uparrow	Text Fidelity \uparrow	Temporal Consistency \uparrow	Overall \uparrow
Tune-A-Video [8]	1.69v.s. 98.31(ours)	11.69v.s. 88.31(ours)	0.39v.s. 99.16(ours)	0.89v.s. 99.11(ours)
FateZero [6]	7.64v.s. 92.36(ours)	17.54v.s. 82.46(ours)	3.62v.s. 96.38(ours)	5.75v.s. 94.25(ours)
CoDeF [5]	2.76v.s. 97.24(ours)	10.18v.s. 89.82(ours)	1.87v.s. 98.13(ours)	3.90v.s. 96.10(ours)
TokenFlow [1]	10.85v.s. 89.15(ours)	26.32v.s. 73.68(ours)	8.19v.s. 91.81(ours)	12.29v.s. 87.71(ours)

2 INTRODUCES ADDITIONAL RESULTS

2.1 Ablation study on the model size of DINOv2

To investigate the influence of the model size of DINOv2[4] on its performance, we conducted ablation experiments on a subset of 8 videos from the HDTF [10] dataset. It is important to mention that the results reported in the original paper were obtained using the smallest DINOv2 architecture, ViT-S/14.

To evaluate the quality of the reconstructions, we employed three widely used metrics: structural similarity index (SSIM), peak signal-to-noise ratio (PSNR), and perceptual image similarity (LPIPS).

Table 2: Ablation study on the model size of DINOv2.

Arch	SSIM (%) \uparrow	PSNR \uparrow	LPIPS \downarrow
ViT-S/14	97.1	33.39	0.029
ViT-B/14	97.6	33.71	0.025
ViT-L/14	97.8	33.83	0.024
ViT-g/14	98.3	34.11	0.021

Table2 presents a comparison of the results obtained with DINOv2 models of different sizes, revealing the impact of model capacity on reconstruction performance. It is evident that as the model size increases, the reconstruction performance improves. This improvement can be attributed to the larger number of parameters in larger models, allowing them to capture more intricate patterns and relationships in videos.

However, it is important to note that larger models also come with increased computational requirements for both training and inference. Therefore, when deciding on the model size, it is crucial to consider the available computational resources and the specific accuracy requirements of video editing task.

By analyzing DINOv2's performance under various model sizes, we gain valuable insights into the trade-off between model complexity and performance. These insights enable researchers and practitioners to make informed decisions when selecting the optimal model size for video editing tasks, striking a balance between accuracy and computational efficiency.

2.2 Comparative preference results from the user study

To gather human preferences, we recruited 77 volunteers who actively participated in our study. We presented them with both our edited results and baseline editing results, accompanied by corresponding textual descriptions. Subsequently, we requested the volunteers to rate the results on a scale of 1 to 5, considering four dimensions: motion consistency, text fidelity, temporal consistency, and overall quality.

Initially, in the original article, we calculated the average scores provided by the volunteers to determine the final results for each assessment metric. However, we acknowledged that each individual had their own subjective criteria for evaluating the results. To address this variability, we opted to calculate relative preferences.

By comparing the scores assigned by each volunteer for the different editing outcomes, we obtained relative preference statistics that offer a more dependable basis for comparing the editing results.

As shown in Table 1, our method, consistently achieved higher scores across all evaluation metrics compared to the other models. Our results indicate superior performance in terms of motion consistency, text fidelity, temporal consistency, and overall quality.

2.3 Comparison of streamability

We conducted streamability tests on a subset of 8 videos extracted from the HDTF dataset [10]. To ensure a fair evaluation, we divided each video into two segments: the first 1000 frames were utilized for training, while the remaining 500 frames were employed for testing. The reconstructed results were then assessed to evaluate the streamability performance.

Table 3: Streamability Performance.

SSIM (%) \uparrow	PSNR \uparrow	LPIPS \downarrow
95.7	31.89	0.041

Measuring streamability for time-dependent models like CoDeF [5] and Atlas [3] Networks can indeed be challenging. As shown in Table 3, the quality of the reconstruction results tends to degrade compared to the first 1000 frames. However, even under such circumstances, our method demonstrates superior performance with remarkable streamability capabilities.

3 DISCUSSES THE LIMITATIONS OF OUR APPROACH

Two notable limitations are related to the slow speed of DINOv2 [4] feature extraction and the challenges associated with controlnet's bias in generating individuals with their eyes glued to the screen.

The first limitation lies in the slow speed of DINOv2 [4] feature extraction, which can significantly impact the real-time editing performance. The computational complexity of extracting features using DINOv2 [4] may hinder the ability to generate feature representations quickly enough for real-time editing applications. This limitation restricts the system's responsiveness and may not be suitable for scenarios that require fast editing, such as interactive applications or live events.

The second limitation arises from ControlNet's [9] bias towards generating individuals with their eyes glued to the screen. This bias can make it challenging to achieve accurate eye alignment, leading to a mismatch of eyes in the editing results. This limitation adversely affects the visual realism and quality of the rendered individuals.

REFERENCES

- [1] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. 2023. TokenFlow: Consistent Diffusion Features for Consistent Video Editing. *arXiv preprint arxiv:2307.10373* (2023).

- [2] Ivan Grishchenko, Artsiom Ablavatski, Yury Kartynnik, Karthik Raveendran, and Matthias Grundmann. 2020. Attention mesh: High-fidelity face mesh prediction in real-time. *arXiv preprint arXiv:2006.10962* (2020).
- [3] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. 2021. Layered neural atlases for consistent video editing. *ACRM Trans. Graph.* 40, 6 (2021), 1–12.
- [4] Maxime Oquab, Timothée Darcet, Theo Moutakanni, and et al. 2023. DINOv2: Learning Robust Visual Features without Supervision. *arXiv:2304.07193* (2023).
- [5] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. 2023. CoDeF: Content deformation fields for temporally consistent video processing. *arXiv preprint arXiv:2308.07926* (2023).
- [6] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. 2023. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv preprint arXiv:2303.09535* (2023).
- [7] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. 2021. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*. 12179–12188.
- [8] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2022. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *arXiv preprint arXiv:2212.11565* (2022).
- [9] Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543* (2023).
- [10] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. 2021. Flow-Guided One-Shot Talking Face Generation With a High-Resolution Audio-Visual Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3661–3670.

233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290

291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348