COUNT TO FIVE! EFFICIENT TEST-TIME PERSONALIZATION FOR POSE-GUIDED HUMAN VIDEO SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pose-driven human video generation aims to synthesize realistic videos of a person following a specified pose sequence. Existing methods rely on static models that generalize poorly to unseen subjects, leading to appearance drift and mismatched details. We study test-time personalization for pose-guided video diffusion, where a few posed snapshots of the target subject are available at inference. Naive fine-tuning on these snapshots is slow and prone to overfitting, and initial updates can degrade the pretrained model. We therefore propose AdaptAnyone, to our knowledge the first meta-learned framework for test-time personalization of pose-guided video diffusion models. Our instance-level meta-learning strategy treats each training subject as a separate task and learns a parameter-efficient initialization for personalizing a new subject in a few gradient steps. Guided by a sensitivity analysis, we efficiently adapt the components that most influence subject appearance and temporal consistency. With three gradient steps, AdaptAnyone outperforms both the base model and naive fine-tuning, which requires orders of magnitude more steps, in subject fidelity, image quality, and temporal stability.

1 INTRODUCTION

Pose-driven human video generation (Liao et al., 2024) aims to synthesize realistic videos of a person following a specified sequence of body poses, given one or a few reference images of that person. This task has broad applications in entertainment, content creation, and digital avatars. Beyond visual realism and temporal coherence, a key requirement for practical deployment is to preserve the holistic appearance of the human subject throughout the generated video. This challenge is amplified in cross-domain settings, where the subject, background, and motion style may all differ from training data, making per-instance adaptation essential for reliable deployment.

Built on off-the-shelf video diffusion models, recent methods have achieved strong pose following and temporal consistency (Kim et al., 2024; Karras et al., 2023). Most efforts introduce improved conditioning to inject pose and reference imagery for stronger identity control. Some works (Tu et al., 2025; Xu et al., 2024; Liu et al., 2025) add task-specific face or motion refinement modules to improve controllability. However, these approaches cannot generalize well to unseen subjects and largely under-address personalization. Specifically, once trained, the model is fixed and remains static: every new subject is handled by the same generic model at inference. Such approaches (Xue et al., 2025; He et al., 2025) often yield characters that only loosely resemble the target, with mismatched facial features, clothing, and body details, and appearance that drifts over time.

Therefore, in this paper, we study test-time adaptation for pose-guided video diffusion, where a small set of posed snapshots of the target subject, readily obtainable in practice, e.g., from a short clip of the subject, is leveraged at inference for personalization. The most straightforward approach is to fine-tune the pre-trained model directly on these snapshots at inference time, which we term Naive Test-Time Adaptation (Naive TTA). However, this is costly and prone to overfitting, degrading both efficiency and the underlying generative prior. Similar challenges have been identified in the personalization of image generation models (Ruiz et al., 2023; 2024; Zeng et al., 2024).

To address this, we exploit a learn-to-adapt strategy and therefore propose *AdaptAnyone* (Figure 1), a test-time personalization framework via learned adaptation for rapid and reliable personalization

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

of the pose-guided human video diffusion model. Specifically, we develop an instance-level learn-to-adapt approach where each human subject is treated as a task. During training, we simulate test-time adaptation and design the training objective to ensure the simulated adaptation benefits video generation for unseen poses. This enables the model to learn a parameter-efficient adaptation prior. A perturbation-based sensitivity analysis further shows that subject appearance and temporal consistency depend on a small subset of transformer components; restricting adaptation to this subset outperforms adapting LoRA in all blocks (Section 5.4). Therefore, during inference, the pre-trained model can be adapted to a new subject in just three gradient steps, given a few snapshots of the person. Our main contributions are:

- To our knowledge, we are the first to meta-learn test-time personalization for pose-guided *video diffusion* models.
- We exploit a learn-to-adapt strategy and propose a *model-agnostic* test-time personalization framework that learns a parameter-efficient initialization for fast, reliable adaptation from a few frames of an unseen subject, together with a sensitivity analysis that identifies which components of the video diffusion transformer to adapt.
- With three gradient steps, AdaptAnyone outperforms the base model and Naive TTA, which requires orders of magnitude more steps, and achieves the best results on nearly all benchmark metrics.

2 Related Works

2.1 Pose-Guided Human Animation and Video Synthesis

Pose-driven human animation has evolved from early GAN-based methods (Ma et al., 2017; Siarohin et al., 2019; Wang et al., 2018; Liu et al., 2019), which often exhibited texture distortions and identity drift, to diffusion-based frameworks. The adoption of diffusion models, starting with text-to-video systems (Ho et al., 2022; Singer et al., 2023; Blattmann et al., 2023) as a foundation, led to dedicated pose-controlled models (Hu et al., 2024; Zhang et al., 2025b; Wang et al., 2025a; Xu et al., 2024; Wang et al., 2024b) that substantially outperformed GANs. Recent works further explore improving consistency via 3D priors (Weng et al., 2022; Zhang et al., 2025a; Song et al., 2023) or attention-based spatial alignment mechanisms (Xu et al., 2024).

Despite these advancements, these models remain static once trained and often fail to preserve an unseen subject's appearance (Xue et al., 2025). Image-domain methods improve identity consistency by refining face embeddings (Li et al., 2024; Wang et al., 2024c; Yan et al., 2023), generating facial prompts (Huang et al., 2026), or contrastive alignment losses (Guo et al., 2024); unlike these face-centric approaches, our method targets holistic subject appearance.

Other works explore personalized diffusion to address these pitfalls (Ruiz et al., 2023), including video customization methods such as DreamVideo (Wei et al., 2024) that fine-tune separate identity and motion adapters per subject, but they require many optimization steps or extensive reference images. Feed-forward alternatives (Ye et al., 2023; Li et al., 2023) bypass per-subject optimization by encoding identity via pre-trained image encoders, yet their fixed representations often lack the fidelity of optimization-based methods for fine-grained identity details in video.

2.2 Test-Time Adaptation and Meta-Learning

Test-time adaptation (TTA) adapts models to novel input distributions at inference, typically for classification and segmentation via entropy minimization and self-training (Wang et al., 2021; Liu et al., 2021). However, these methods handle distribution shifts (e.g., style or corruption), whereas our task requires adapting to a particular individual's appearance. In generative modeling, personalization has been explored through fine-tuning (Gal et al., 2023a; Ruiz et al., 2023), parameter-efficient adaptation (Hu et al., 2022; Tewel et al., 2023), latent inversion (Roich et al., 2022), and editing with minimal samples (Brooks et al., 2023). To reduce per-subject optimization cost, encoder-based methods such as Gal et al. (2023b) predict initializations requiring only a few fine-tuning steps, while Ruiz et al. (2024) train a hypernetwork to directly predict LoRA weights for fast refinement. Most recently, Topal et al. (2025) meta-learn a domain-specific LoRA prior for identity personalization of image diffusion models. However, these methods target *image* generation and do not address

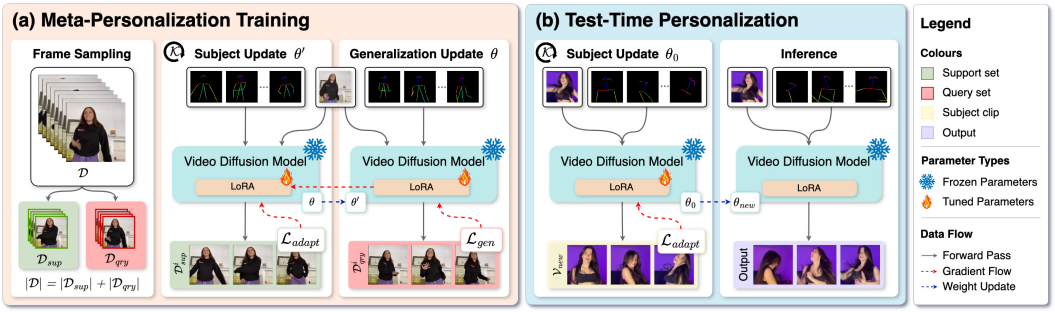

Figure 1: **Overview of AdaptAnyone.** Given a few posed frames of an unseen subject at test time, AdaptAnyone personalizes a pose-guided video diffusion model in $\mathcal{K}$=3 gradient steps. (a) **Meta-Personalization.** Each training clip is treated as a task $\mathcal{T}$, split into a support set $\mathcal{D}_{\text{sup}}$ and query set $\mathcal{D}_{\text{qry}}$. *Subject adaptation:* starting from a global initialization θ, we adapt only lightweight LoRA parameters on $\mathcal{D}_{\text{sup}}$ for $\mathcal{K}$ steps, yielding task-specific weights θ' (backbone frozen). *Generalization loop:* we evaluate θ' on $\mathcal{D}_{\text{qry}}$ and update θ with the resulting first-order gradient, producing the adaptation prior θ_0. (b) **Test-time personalization (inference).** From θ_0, we run the same $\mathcal{K}$-step subject adaptation on the reference frames to obtain θ_{new}, then generate videos under arbitrary poses.

the temporal consistency of video. In contrast, rather than accelerating image-level personalization, we formulate instance-level meta-learning directly over a video diffusion backbone.

This builds on meta-learning, which learns initializations that generalize to unseen tasks from few examples (Finn et al., 2017; Rajeswaran et al., 2019). In computer vision, meta-learning has been applied to few-shot recognition (Lu et al., 2020; Snell et al., 2017; Sung et al., 2018), video frame interpolation (Choi et al., 2020), video stabilization (Ali et al., 2024), and long-form video generation (Dalal et al., 2025). Meta-learned personalization has been explored for talking-head synthesis (Zakharov et al., 2019; Zhang et al., 2023) and for pose-guided retargeting with frame-level generators (MetaPix; Lee et al., 2020), while few-shot vid2vid (Wang et al., 2019) predicts generator weights from example frames. In contrast, AdaptAnyone meta-learns LoRA parameters at sensitivity-selected components of a video diffusion transformer, optimizes their initialization with a support/query objective on held-out poses, and personalizes in three gradient steps.

3 Preliminary: Pose-Guided Video Diffusion

We build our framework upon a pre-trained, pose-guided video generation model v_Φ (Hu et al., 2024; Wang et al., 2025a); we use "video diffusion model" for both diffusion and flow-matching models (Lipman et al., 2023). A VAE $\mathcal{E}_{vae}$ encodes a clip V into a latent $\mathbf{z}_0$, and a DiT v_Φ (Peebles & Xie, 2023) is conditioned on $\mathcal{C}$: a text prompt c_{text}, a reference image embedding c_{ref}, and target poses c_{pose} (e.g., from DW-Pose (Yang et al., 2023)). UniAnimate-DiT (Wan2.1 (Wan et al., 2025)) is trained with flow matching: with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $t \in [0, 1]$, and $\mathbf{z}_t = (1-t)\,\mathbf{z}_0 + t\,\boldsymbol{\epsilon}$, v_Φ regresses the velocity $\boldsymbol{\epsilon} - \mathbf{z}_0$:

$$\mathcal{L}_{simple} = \mathbb{E}_{\mathbf{z}_0,t,\boldsymbol{\epsilon},\mathcal{C}}\left[\left\|(\boldsymbol{\epsilon} - \mathbf{z}_0) - v_\Phi(\mathbf{z}_t, t, c_{text}, c_{ref}, c_{pose})\right\|^2\right]. \tag{1}$$

DynamiCtrl (CogVideoX (Yang et al., 2025)) is instead a v-prediction diffusion model; $\mathcal{L}_{simple}$ denotes each backbone's native objective, and sampling maps noise to a new latent $\hat{\mathbf{z}}_0$.

4 Our Approach: AdaptAnyone

Our goal is to enable a large, pre-trained video diffusion model to rapidly personalize to a new human subject from a few snapshots of that person given at test time (Figure 1).

4.1 Motivation

The most direct approach to test-time adaptation is to fine-tune the pre-trained pose-guided video diffusion model with its training objective $\mathcal{L}_{simple}$ applied to the image frames provided at inference.

This naive strategy has two limitations. First, because the optimization both trains and evaluates on the same few reference frames, there is no learning signal that encourages the model to disentangle the subject's transferable appearance from frame-specific context. Consequently, the model *memorizes* the specific background, lighting, and outfit present in the frames rather than *learning* a flexible, pose-robust subject representation. Second, the aggressive gradient updates required for adaptation cause *catastrophic forgetting*: they degrade the pretrained generative prior on which visual quality and temporal coherence depend; with up to 200 steps, Naive TTA underperforms the unadapted base model on four of six metrics (Section 5.3). Moreover, Naive TTA remains below our three-step adaptation on all metrics even after 1500 gradient steps per subject (Section 5.3), which is *time-consuming* and *impractical* for interactive use.

To address this, we train the model to *learn how to adapt* robustly.

4.2 Learning to Personalize via Subject-Level Adaptation

Inspired by the meta-learning strategy for fast adaptation (Finn et al., 2017; Nichol et al., 2018), where a model is optimized over a distribution of tasks to learn a shared initialization that generalizes with few examples, we introduce an instance-level personalization framework tailored for pose-guided video diffusion. Specifically, we treat each human subject as a distinct personalization task.

Learning Appearance via Subset Adaptation. Adapting the entire billion-parameter backbone Φ is resource-intensive. Following parameter-efficient methods such as LoRA (Hu et al., 2022; Zhao et al., 2024; Xing et al., 2024), we hypothesize that a subject's appearance can be captured by a small set of parameters: we freeze Φ to preserve its generative prior and update only θ ($\theta \ll \Phi$), the LoRA layers in the DiT blocks, yielding $v_{\Phi,\theta}(\cdot)$.

Sensitivity-Guided LoRA Placement. Injecting LoRA into all components of all blocks is suboptimal, as many components contribute little to adaptation quality while adding parameters. To identify the most impactful subset, we perform a perturbation-based sensitivity analysis: for each component, i.e., each self-attention, cross-attention, or feed-forward layer of a DiT block, we apply a small random perturbation to its LoRA weights and measure the resulting change in two latent-space proxy objectives: frame-level reconstruction error (appearance quality) and inter-frame prediction variance (temporal consistency). We then rank all components by their combined sensitivity and retain only the top-scoring subset: 29 of the 120 components of UniAnimate-DiT's 40 blocks (Figure 4; Appendix A.2). This targeted placement restricts adaptation to the most sensitive components and leaves all other pretrained weights unchanged.

Subject-Task Formulation. During meta-personalization training, a task $\mathcal{T}$ is a video V_i of one person from $\mathcal{D}_{train}$, split into a *support set* $\mathcal{D}_{sup}$ of $\mathcal{M}$ posed frames, which simulates the test-time reference frames, and a *query set* $\mathcal{D}_{qry}$ of held-out frames used to evaluate the personalization.

Personalization Objective. Meta-personalization training (Algorithm 1) is a bi-level optimization: a subject-adaptation loop simulates personalization, and a generalization loop updates the model.

Subject Adaptation Loop (Simulated Adaptation): Starting from the current prior θ, we simulate test-time adaptation by minimizing $\mathcal{L}_{simple}$ on $\mathcal{D}_{sup}$, encoding the subject's appearance in θ':

$$\theta' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{simple}(v_{\Phi,\theta}, \mathcal{D}_{sup}), \tag{2}$$

where α is the subject adaptation learning rate and $\mathcal{L}_{simple}$ is defined in Eq. (1).

Generalization Loop (Generalization Update): The generalization loop evaluates θ' on the query set $\mathcal{D}_{qry}$ of the same subject, with $\mathcal{L}_{simple}$ as the generalization loss $\mathcal{L}_{gen}$. Following first-order MAML (FOMAML) (Nichol et al., 2018), we compute $\nabla_\theta \mathcal{L}_{\text{gen}}$ at θ' without backpropagating through the inner steps, which keeps the outer loop tractable for billion-parameter backbones while still optimizing for generalization to held-out frames.

$$\mathcal{L}_{gen} \leftarrow \mathcal{L}_{simple}(v_{\Phi,\theta'}, \mathcal{D}_{qry}), \qquad \theta \leftarrow \theta - \beta \nabla_{\theta'} \mathcal{L}_{gen}, \tag{3}$$

where β is the generalization loop learning rate; in practice, this update uses Adam.

Algorithm 1 AdaptAnyone meta-personalization training

Input: Training dataset $\mathcal{D}_{train}$
Output: Adapted parameter θ_0
Pre-trained backbone: frozen video diffusion backbone Φ
Trainable parameters: LoRA weights θ
Hyperparameters: subject adaptation loop learning rate α; generalization loop learning rate β; number of subject adaptation loop steps $\mathcal{K}$; total meta personalization training steps T

Initialize θ (adaptable parameters)
for $t = 1$ to T **do** ▷ generalization loop over training-iterations
  Sample task $\mathcal{T} \sim \mathcal{D}_{train}$
  Split $\mathcal{T}$ into support $\mathcal{D}_{sup}$ and query $\mathcal{D}_{qry}$
  $\theta' \leftarrow \theta$ ▷ Initialize task-specific parameters
  ▷ **— subject adaptation loop (Adaptation) —**
  for $k = 1$ to $\mathcal{K}$ **do**
    $\mathcal{L}_{adapt} \leftarrow \mathcal{L}_{simple}(v_{\Phi,\theta'}, \mathcal{D}_{sup})$
    ▷ Update θ' with standard GD
    $\theta' \leftarrow \theta' - \alpha \nabla_{\theta'} \mathcal{L}_{adapt}$
  end for
  ▷ **— generalization loop (Generalization Update) —**
  ▷ Evaluate adapted θ' on $\mathcal{D}_{qry}$
  $\mathcal{L}_{gen} \leftarrow \mathcal{L}_{simple}(v_{\Phi,\theta'}, \mathcal{D}_{qry})$
  $\theta \leftarrow \theta - \beta \nabla_{\theta'} \mathcal{L}_{gen}$ ▷ Update the adaptation prior
end for
return $\theta_0 \leftarrow \theta$ ▷ The optimal initialization

4.3 Fast Few-Shot Personalization

At test time (Algorithm 2, Appendix A), we load the frozen backbone Φ and the adaptation prior θ_0, run the same $\mathcal{K}$-step subject adaptation on the $\mathcal{M}$ reference frames of V_{new} to obtain θ_{new}, and generate videos from a single reference image under any target pose sequence c_{pose} with $v_{\Phi,\theta_{new}}$.

5 Experiments

5.1 Experimental Setup

Table 1: Comparison on TikTok (Jafarian & Park, 2021) and Fashion (Zablotskaia et al., 2019) (cells: TikTok/Fashion). Competitor numbers are published (–: not reported); DynamiCtrl rows use our re-trained checkpoint (Appendix D.2). Ours (1 frame) uses one reference image, like the baselines.

Model	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	FID-VID↓
MusePose (Tong et al., 2024)	18.20/22.20	0.757/0.896	0.248/0.067	41.99/14.95	532.75/96.17	14.60/10.94
Moore-AA (MooreThreads, 2024)	18.62/20.83	0.764/0.880	0.230/0.093	37.28/27.74	501.22/149.66	12.40/10.13
ControlNeXt (Peng et al., 2024)	16.31/18.48	0.728/0.853	0.296/0.132	33.48/13.82	548.01/143.02	28.23/14.60
Animate-X (Tan et al., 2025)	16.71/22.15	0.743/0.893	0.285/0.069	32.77/10.11	508.63/70.47	17.47/11.87
MimicMotion (Zhang et al., 2025b)	19.30/23.80	0.751/0.913	0.220/0.061	34.88/15.40	472.51/80.89	9.30/8.17
RealisDance-DiT (Zhou et al., 2025)	17.55/23.33	0.717/0.908	0.261/0.053	30.39/10.81	458.81/72.94	–/–
MTVCraft (Ding et al., 2026)	19.84/23.90	0.779/0.923	**0.217**/0.057	20.70/8.74	276.65/64.88	**7.31**/4.41
DynamiCtrl: Base (Zhao et al., 2025)	18.26/19.41	0.804/0.892	0.247/0.070	31.21/26.91	502.35/144.01	25.62/9.16
DynamiCtrl: Ours	18.85/21.23	0.827/0.925	0.219/0.064	28.84/21.33	349.21/98.23	17.21/8.19
UniAnimate-DiT: Base (Wang et al., 2025b)	19.35/23.52	0.765/0.907	0.235/0.060	28.47/12.79	402.14/88.36	9.12/6.12
UniAnimate-DiT: Ours (1 frame)	19.48/23.94	0.784/0.925	0.229/0.053	25.31/8.77	341.28/63.93	8.94/5.21
UniAnimate-DiT: Ours (3 frames)	**19.97/24.12**	**0.832/0.961**	**0.217/0.051**	**19.07/8.72**	**178.86/60.19**	7.89/**4.08**

Datasets. For meta-personalization training (Section 4.2), we use TikTok (Jafarian & Park, 2021) and UBC Fashion (Zablotskaia et al., 2019), with their train and test splits for in-domain evaluation. For cross-domain evaluation, we curate *Unseen100-Val*, a set of 100 in-the-wild videos spanning real and animated subjects with diverse motion, lighting, and backgrounds, disjoint from all training data and evaluated zero-shot (details in Appendix D).

Base Models. To demonstrate model-agnosticism, we apply our method to two base models: **UniAnimate-DiT** (Wang et al., 2025b), based on Wan2.1 (Wan et al., 2025) 14B backbone, and

DynamiCtrl (Zhao et al., 2025), based on CogVideoX (Yang et al., 2025) 5B. In both, the backbone Φ is frozen and only LoRA adapters θ are trained; for UniAnimate-DiT, θ is its pretrained LoRA, attached to every DiT block (Appendix A.2).

Table 2: Cross-domain comparison on Unseen100-Val after training on TikTok. All methods are run by us with identical clips and code; MTVCraft is omitted (it requires SMPL-based 4D motion).

Model	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	FID-VID↓
MusePose (Tong et al., 2024)	16.92	0.748	0.268	41.82	531.47	16.91
Moore-AA (MooreThreads, 2024)	16.38	0.721	0.284	42.15	567.83	16.42
ControlNeXt (Peng et al., 2024)	14.58	0.647	0.341	51.73	692.14	23.87
Animate-X (Tan et al., 2025)	15.74	0.718	0.279	<u>39.64</u>	598.71	17.15
MimicMotion (Zhang et al., 2025b)	15.87	0.709	0.293	45.28	618.35	17.93
RealisDance-DiT (Zhou et al., 2025)	16.14	0.731	0.274	40.51	548.29	16.52
DynamiCtrl: Base (Zhao et al., 2025)	17.56	0.781	0.263	46.67	498.24	17.68
DynamiCtrl: Ours	<u>17.89</u>	<u>0.804</u>	<u>0.245</u>	42.17	<u>398.92</u>	<u>16.03</u>
UniAnimate-DiT: Base (Wang et al., 2025b)	17.65	0.785	0.246	43.21	502.52	16.29
UniAnimate-DiT: Ours	**18.21**	**0.814**	**0.239**	**39.23**	**325.65**	**15.56**

Implementation Details. For meta-personalization training, we use Adam (Kingma & Ba, 2015) for the generalization loop (β=10^{-5}) and plain gradient descent for the subject adaptation loop (α=10^{-4}; 3×10^{-4} for DynamiCtrl), LoRA rank and scaling factor 32 (16 for DynamiCtrl), and 10,000 generalization-loop steps (batch size 1) on 4× NVIDIA H100-80G GPUs, at each base model's native resolution (Appendix A).

For our main experiments, we use $\mathcal{M}$=3 support frames, reflecting a realistic scenario where users provide a short clip of the subject, and $\mathcal{K}$=3 subject adaptation steps. Support frames are selected automatically from a held-out segment of each test video that is never evaluated, and one of them serves as the reference image (Appendix D.2). As in standard self-driven evaluation, each method's reference and evaluated frames come from the same video. Since only LoRA parameters change, inference after adaptation has the same cost as the base model; the one-time $\mathcal{K}$ gradient steps take under 5 s on a single GPU for UniAnimate-DiT (7.8 s for DynamiCtrl).

Metrics. We report PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) (VGG) for per-frame reconstruction; FID (Heusel et al., 2017), FID-VID (Balaji et al., 2019), and FVD (Unterthiner et al., 2018) for perceptual quality and temporal consistency (Appendix B); and adaptation time, the wall-clock time of the test-time gradient steps in minutes.

5.2 Comparison against SOTA

Quantitative Results. We compare AdaptAnyone against recent pose-guided human animation methods, including MusePose (Tong et al., 2024), Moore-AnimateAnyone (MooreThreads, 2024), ControlNeXt (Peng et al., 2024), Animate-X (Tan et al., 2025), MimicMotion (Zhang et al., 2025b), RealisDance-DiT (Zhou et al., 2025), and MTVCraft (Ding et al., 2026). With a single frame, the same input the baselines receive, AdaptAnyone outperforms every baseline except MTVCraft on at least five of six TikTok metrics and exceeds MTVCraft on four of six Fashion metrics; three frames improve it further. Section 5.3 isolates our contribution under identical inputs.

As shown in Table 1, our adaptation improves both base models on all metrics on TikTok and UBC Fashion. On UniAnimate-DiT, it reduces FVD by $2.2\times$ and FID by over 30% and achieves the best or tied-best result on 11 of 12 metrics; the remaining FID-VID gap to MTVCraft on TikTok likely reflects its 4D motion tokenization, which is complementary to our framework. DynamiCtrl improves consistently despite its smaller 5B backbone, confirming generalization across architectures. Across three random seeds, AdaptAnyone varies less than the base model on five of six metrics (e.g., FVD standard deviation 26.2 vs. 119.1; Table 11).

Table 2 reports zero-shot transfer to our in-the-wild evaluation set. All methods degrade relative to in-domain performance, but AdaptAnyone retains a clear advantage; its largest gain is on FVD (35% over the base model), suggesting that the meta-learned initialization captures transferable subject appearance rather than dataset-specific priors. Beyond self-driven reconstruction, we also test cross-reenactment on Unseen100-Val, with driving motion from another subject: AdaptAnyone improves FaceID from 0.433 to 0.459, Body-Sim from 0.728 to 0.754, and Pose-Acc from 0.442 to 0.456 over the base model, exceeding 1500-step Naive TTA on all three (Appendix C.3).



Figure 2: **Qualitative comparison against state-of-the-art methods on unseen subjects.** Each column animates a different subject under a target pose sequence. Our 3-step adaptation preserves the subject's holistic appearance (face, clothing, and body proportions) and temporal coherence.

Qualitative Results. Figure 2 shows that competitor methods frequently exhibit appearance drift and temporal artifacts under challenging motion, while AdaptAnyone consistently preserves holistic appearance and temporal coherence. Additional qualitative results are in Appendix E.

5.3 Comparison against Naive TTA

Table 3: **AdaptAnyone vs. Naive TTA at varying fine-tuning budgets.** Our 3-step adaptation outperforms Naive TTA on all metrics at every budget up to 1500 steps (UniAnimate-DiT).

Model	Fine-Tuning Steps	Fine-Tuning time (min)	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	FID-VID↓
Base	0	N/A	19.35	0.765	0.235	28.47	402.14	9.12
Naive TTA	3	0.06	18.70	0.787	0.252	39.41	343.61	11.26
	100	2.06	18.73	0.802	0.239	33.05	303.00	14.81
	200	4.49	18.97	0.805	0.240	32.13	292.63	15.37
	500	10.5	19.01	0.811	0.232	27.61	269.99	14.61
	1000	20.8	19.70	0.817	0.228	25.09	241.27	13.84
	1500	31.1	19.93	0.819	0.220	22.03	219.52	10.33
AdaptAnyone	**3**	**0.06**	**19.97**	**0.832**	**0.217**	**19.07**	**178.86**	**7.89**

We treat direct fine-tuning of LoRA parameters on the support frames without meta-learning as our primary baseline (Naive TTA), since it isolates the effect of our meta-learned initialization (setup in Appendix A.3). With 3 gradient steps (under 5 s), AdaptAnyone outperforms Naive TTA with 1500 steps (over 30 min per subject) on all metrics (Table 3).

With 3–200 steps, Naive TTA performs *worse* than the unadapted base model on PSNR, LPIPS, FID, and FID-VID, indicating that a generic initialization degrades the prior before it helps, and its FID-



Figure 3: **Consecutive frame-difference maps for Naive TTA (3 and 1500 steps) vs. AdaptAnyone.** Naive TTA at 3 steps shows abrupt transitions from insufficient adaptation; at 1500 steps, localized flickering remains. Ours yields low residuals, i.e., stable temporal coherence.

VID is non-monotonic, peaking at 200 steps. Our meta-learned initialization bypasses this unstable regime, as the uniformly low frame-to-frame residuals in Figure 3 corroborate; since AdaptAnyone also attains the highest pose accuracy (Appendix C.3), these residuals reflect reduced flicker rather than suppressed motion. DynamiCtrl results are in Appendix C.2.

5.4 Ablation Study

We ablate on UniAnimate-DiT with the default configuration ($\mathcal{K}$=3, $\mathcal{M}$=3, LoRA rank 32, α=10^{-4}, β=10^{-5}, sensitivity-guided placement) unless stated otherwise. DynamiCtrl results and ablations of the support-frame count, LoRA rank, and learning rates are in Appendices C.2 and C.1.

Table 4: **Component ablation.** Meta-learning gives the largest gain over Naive TTA and outperforms LoRA fine-tuning on the same data. The support/query split improves temporal generalization (FVD), and sensitivity-guided selection improves both per-frame fidelity and distributional quality.

Configuration	PSNR↑	SSIM↑	LPIPS↓	FID↓	FID-VID↓	FVD↓
Base Model	19.35	0.765	0.235	28.47	9.12	402.14
Naive TTA, $\mathcal{K}$=3 (no meta)	18.70	0.787	0.252	39.41	11.26	343.61
LoRA SFT on training data, $\mathcal{K}$=3 (no meta)	19.41	0.782	0.230	26.09	9.11	302.77
FOMAML + LoRA, no query split	19.78	0.823	0.220	21.84	8.61	291.41
FOMAML + LoRA + query split, all blocks	19.60	0.812	0.223	22.11	8.32	262.51
Full AdaptAnyone (+sensitivity)	**19.97**	**0.832**	**0.217**	**19.07**	**7.89**	**178.86**

Component Ablation. Table 4 adds components progressively. Without meta-learning, Naive TTA at $\mathcal{K}$=3 *worsens* FID from 28.47 to 39.41: fine-tuning from the base weights cannot adapt in so few steps. LoRA SFT, which fine-tunes the pretrained LoRA on the same data for the same number of steps without meta-learning, underperforms every meta-learned variant on all six metrics, so additional in-domain training does not explain the gains. Meta-learning yields the largest improvement on five of six metrics and lifts all metrics above the baseline. The support/query split further reduces FVD from 291.41 to 262.51 at a small cost in per-frame metrics, indicating that separating the meta-objective from the adaptation target improves generalization to unseen poses. Meta-learning with all blocks adapted, as in prior meta-learned personalization (Lee et al., 2020; Topal et al., 2025), reaches FVD 262.51, whereas sensitivity-guided selection lowers it to 178.86, below 1500-step Naive TTA (219.52). Without meta-learning, the same selection (Naive TTA) also falls short; both are required.

Subject Adaptation Steps ($\mathcal{K}$). Table 5 varies $\mathcal{K}$, with a separate model meta-trained for each value. Without adaptation ($\mathcal{K}$=0), θ_0 stays close to the base model (FVD 389.26 vs. 402.14), so the gains come from test-time personalization rather than from the meta-trained prior alone. One step already improves over the base model on all metrics except FID-VID; $\mathcal{K}$=3, our default, is best or tied for best on every metric; and $\mathcal{K}$=5 degrades all metrics, indicating overfitting to the support frames.

Table 5: **Ablation on the number of subject adaptation steps ($\mathcal{K}$).** Each $\mathcal{K}\geq 1$ uses a model meta-trained with the same $\mathcal{K}$; $\mathcal{K}=0$ evaluates θ_0 without adaptation. We find $\mathcal{K}=3$ provides the optimal balance, while training with more steps leads to performance degradation.

$\mathcal{K}$	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	FID-VID↓
–	19.35	0.765	0.235	28.47	402.14	9.12
0 (θ_0)	19.44	0.779	0.239	26.43	389.26	9.02
1	19.53	0.815	0.219	22.14	184.90	9.15
2	19.65	0.820	**0.217**	20.52	218.40	8.59
3	**19.97**	**0.832**	**0.217**	**19.07**	**178.86**	**7.89**
5	19.39	0.802	0.223	21.30	203.90	8.82

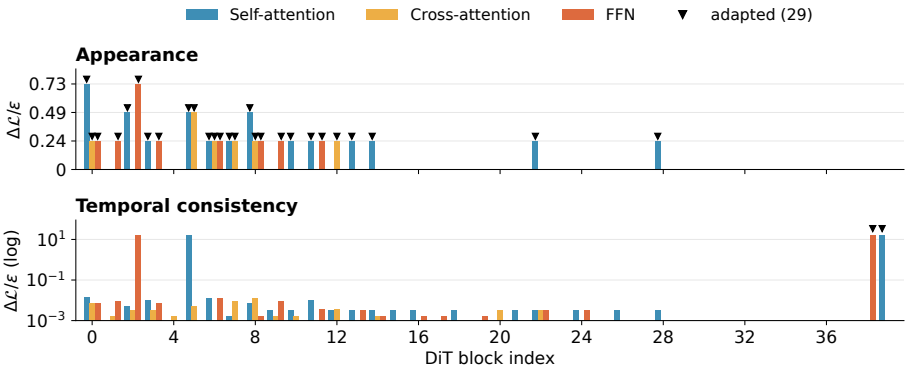

Figure 4: **Sensitivity-guided LoRA placement.** Finite-difference sensitivity ($\Delta\mathcal{L}/\epsilon$) of every component of the 40 UniAnimate-DiT blocks to a Gaussian perturbation of its LoRA weights, for appearance (top) and temporal consistency (bottom, log scale; details in Appendix A.2). Appearance sensitivity is concentrated in blocks 0–14, and temporal sensitivity peaks in early (2, 5) and final (38, 39) blocks. We adapt the 29 components with the highest combined score (▼).

Table 6: **Ablation on LoRA placement strategy.** We compare heuristic block selection with our sensitivity-guided approach. Ours is best on all metrics with only 29 adapted components.

Strategy	Adapted components	PSNR↑	SSIM↑	LPIPS↓	FID↓	FID-VID↓	FVD↓
Base Model (no adaptation)	0	19.35	0.765	0.235	28.47	9.12	402.14
All blocks	120	19.60	0.812	0.223	22.11	8.32	262.51
First 20 blocks	60	19.52	0.801	0.226	24.23	8.67	360.19
Last 20 blocks	60	19.21	0.772	0.237	27.41	9.12	385.21
Sensitivity-Guided	29	**19.97**	**0.832**	**0.217**	**19.07**	**7.89**	**178.86**

LoRA Placement Strategy. Table 6 validates sensitivity-guided placement. Adapting all 40 blocks improves over the base but remains far below ours. First-20 yields reasonable fidelity but poor FVD (360.19), and last-20 changes little relative to the base; adapting fewer blocks is thus not beneficial per se, and performance depends on which components are adapted. The selected components concentrate in blocks 0–14 for appearance and include blocks 38–39 for temporal coherence (Figure 4), achieving the best results while adapting only 24% of the components, the fewest among all strategies.

6 Conclusion

We presented AdaptAnyone, a test-time personalization framework for pose-guided human video diffusion. By treating each subject as a personalization task and learning a parameter-efficient initialization, our method adapts to a new identity in just three gradient steps, achieving substantial gains in subject fidelity, visual quality, and temporal stability over static baselines and naive fine-tuning. Since our framework is task-agnostic by design, it could extend to other conditional generation settings such as style transfer and 3D generation. Its main limitation, sensitivity to low-quality reference frames (e.g., motion blur or occlusion), motivates adaptive frame selection as future work.

AI USE STATEMENT

The research ideas, the method, the design of the experiments, and the analysis and interpretation of the results are the authors' own. We did not use generative AI tools to generate or process data, to formulate the method or its claims, or to implement the method. We used generative AI tools to help draft and edit the manuscript (wording, condensation, LaTeX formatting, reference formatting, and consistency checks), to give feedback on drafts of the paper, and to help find related work and verify citation details against the original sources. We also used them to assist with experiment management on our compute cluster, such as submitting and monitoring jobs and retrieving their logged outputs. We have reviewed all AI-assisted work, and we take full responsibility for the final content of this paper, including any text or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

AdaptAnyone lowers the cost of personalizing a human video model to a new subject, which benefits content creation, accessibility, and avatar applications but also warrants a discussion of misuse. The ease of synthesizing realistic video of a person from a few images raises the risk of non-consensual or deceptive media, such as impersonation or misinformation. We condemn any use that depicts an individual without their explicit consent, and we intend any release to be governed by terms that prohibit such use. Like all generative models, ours can inherit biases from its training data; if the underlying datasets under-represent particular skin tones, body types, ages, or styles of dress, personalization quality may be uneven across groups. Unseen100-Val is curated to cover a broad range of appearances and conditions (Appendix D.1) to support such evaluation; it is compiled from publicly available online videos and used solely for non-commercial research evaluation. We support provenance and detection measures, including watermarking of generated video and research on detecting synthesized motion. Because adaptation operates on a few reference frames of a specific subject, the pipeline is also compatible with consent-based safeguards that verify the subject of those frames before adaptation. The user study (Appendix F) involved voluntary participants who gave consent and only rated generated videos.

REPRODUCIBILITY STATEMENT

The meta-personalization training and test-time adaptation procedures are specified in Section 4, with pseudocode in Algorithms 1 and 2. Appendix A details both base-model architectures, the sensitivity-guided LoRA placement and the selected components for each backbone, and all training hyperparameters, task sampling, and compute requirements (Table 7). Metric definitions are given in Appendix B, and the evaluation protocol (sampler settings, prompt, resolution, support-frame selection, and how each baseline was obtained, including our re-training of DynamiCtrl) is described in Appendix D.2. TikTok and UBC Fashion are used with their standard splits; the curation criteria, composition, and preprocessing of Unseen100-Val are documented in Appendix D.1. Sensitivity to the main hyperparameters is reported in Section 5.4 and Appendix C, and the user-study protocol in Appendix F.

REFERENCES

Muhammad Kashif Ali, Eun Woo Im, Dongjin Kim, and Tae Hyun Kim. Harnessing meta-learning for improving full-frame video stabilization. In *CVPR*, 2024.

Yogesh Balaji, Martin Renqiang Min, Bing Bai, Rama Chellappa, and Hans Peter Graf. Conditional GAN with discriminative filter generation for text-to-video synthesis. In *IJCAI*, 2019.

Andreas Blattmann, Robin Rombach, Hengyuan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Patrick Esser. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, 2023.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CVPR*, 2017.

Jinsun Choi, Hyeongwoo Kim, Jinsun Park, Jung-Il Park, and Seoung Wug Oh. Scene-adaptive video frame interpolation via meta-learning. In *CVPR*, 2020.

Karan Dalal, Daniel Koceja, Gashon Hussein, Jiarui Xu, Yue Zhao, Youjin Song, Shihao Han, Ka Chun Cheung, Jan Kautz, Carlos Guestrin, et al. One-minute video generation with test-time training. In *CVPR*, 2025.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

Yanbo Ding, Xirui Hu, Zhizhi Guo, Yan Zhang, Xinrui Wang, Zhixiang He, Chi Zhang, Yali Wang, and Xuelong Li. MTVCraft: Tokenizing 4D motion for arbitrary character animation. In *ICLR*, 2026.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Daniel Cohen-Or, and Dani Lischinski. An Image is Worth One Word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2023a.

Rinon Gal, Moab Arar, Yuval Atzmon, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Encoder-based domain tuning for fast personalization of text-to-image models. *TOG*, 2023b.

Zinan Guo, Yanze Wu, Zhuowei Chen, Lang Chen, Peng Zhang, and Qian He. PuLID: Pure and lightning ID customization via contrastive alignment. In *NeurIPS*, 2024.

Jingxuan He, Busheng Su, and Finn Wong. PoseGen: In-context LoRA finetuning for pose-controllable long human video generation. *arXiv preprint arXiv:2508.05091*, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NeurIPS*, 2017.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Rui Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen Video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Alain Hore and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. In *ICPR*, 2010.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate Anyone: Consistent and controllable image-to-video synthesis for character animation. In *CVPR*, 2024.

Jiehui Huang, Xiao Dong, Wenhui Song, Zheng Chong, Zhenchao Tang, Jun Zhou, Yuhao Cheng, Long Chen, Hanhui Li, Yiqiang Yan, et al. ConsistentID: Portrait generation with multimodal fine-grained identity preserving. *TPAMI*, 2026.

Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *CVPR*, 2021.

Johanna Karras, Aleksander Holynski, Ting-Chun Wang, and Ira Kemelmacher-Shlizerman. DreamPose: Fashion image-to-video synthesis via stable diffusion. In *ICCV*, 2023.

Jeongho Kim, Min-Jung Kim, Junsoo Lee, and Jaegul Choo. TCAN: Animating human images with temporally consistent pose guidance using diffusion models. In *ECCV*, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Jessica Lee, Deva Ramanan, and Rohit Girdhar. MetaPix: Few-shot video retargeting. In *ICLR*, 2020.

Dongxu Li, Junnan Li, and Steven C. H. Hoi. BLIP-Diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *NeurIPS*, 2023.

Zhen Li, Mingdeng Cao, Xintao Wang, Zhongang Qi, Ming-Ming Cheng, and Ying Shan. PhotoMaker: Customizing realistic human photos via stacked id embedding. In *CVPR*, 2024.

Fangjian Liao, Xingxing Zou, and Waikeung Wong. Appearance and pose-guided human generation: A survey. *ACM Computing Surveys*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *ICLR*, 2023.

Mushui Liu, Dong She, Jingxuan Pang, Qihan Huang, Jiacheng Ying, Wanggui He, Yuanlei Hou, and Siming Fu. TFCustom: Customized image generation with time-aware frequency feature guidance. In *CVPR*, 2025.

Wen Liu, Zhiling Piao, Jie Luo, Yujun Li, Lin Lin, Xiaogang Han, and Shenghua He. Liquid Warping GAN: A unified framework for human image synthesis. In *ICCV*, 2019.

Yue Liu, Yifei Li, Ju Liu, Guan-Horng Li, and Furong Liu. TTT++: When does self-supervised test-time training fail or thrive? In *NeurIPS*, 2021.

Yiwei Lu, Frank Yu, Mahesh Kumar Krishna Reddy, and Yang Wang. Few-shot scene-adaptive anomaly detection. In *ECCV*, 2020.

Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NeurIPS*, 2017.

MooreThreads. Moore-AnimateAnyone. `https://github.com/MooreThreads/Moore-AnimateAnyone`, 2024.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Bohao Peng, Jian Wang, Yuechen Zhang, Wenbo Li, Ming-Chang Yang, and Jiaya Jia. ControlNeXt: Powerful and efficient control for image and video generation. *arXiv preprint arXiv:2408.06070*, 2024.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *NeurIPS*, 2019.

Daniel Roich, Ron Mokady, Or Patashnik, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *TOG*, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. HyperDreamBooth: Hypernetworks for fast personalization of text-to-image models. In *CVPR*, 2024.

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-A-Video: Text-to-video generation without text-video data. In *ICLR*, 2023.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

Guoxian Song, Hongyi Xu, Xiaochen Zhao, You Xie, Tianpei Gu, Zenan Li, Chenxu Zhang, and Linjie Luo. X-UniMotion: Animating human images with expressive, unified and identity-agnostic motion latents. In *SIGGRAPH Asia*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Liangchen Song, Anpei Chen, Zhong Li, Chen Zhang, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. NeRFPlayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE TVCG*, 2023.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to Compare: Relation network for few-shot learning. In *CVPR*, 2018.

Shuai Tan, Biao Gong, Xiang Wang, Shiwei Zhang, Dandan Zheng, Ruobing Zheng, Kecheng Zheng, Jingdong Chen, and Ming Yang. Animate-X: Universal character image animation with enhanced motion representation. In *ICLR*, 2025.

Yoad Tewel, Rinon Gal, Gal Chechik, and Yuval Atzmon. Key-locked rank one editing for text-to-image personalization. In *SIGGRAPH*, 2023.

Zhengyan Tong, Chao Li, Zhaokang Chen, Bin Wu, and Wenjiang Zhou. MusePose: a pose-driven image-to-video framework for virtual human generation. `https://github.com/TMElyralab/MusePose`, 2024.

Barış Batuhan Topal, Umut Özyurt, Zafer Doğan Budak, and Ramazan Gokberk Cinbis. Meta-LoRA: Meta-learning LoRA components for domain-aware ID personalization, 2025.

Shuyuan Tu, Zhen Xing, Xintong Han, Zhi-Qi Cheng, Qi Dai, Chong Luo, and Zuxuan Wu. StableAnimator: High-quality identity-preserving human image animation. In *CVPR*, 2025.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. WAN: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution, 2024a. URL https://arxiv.org/abs/2409.12191.

Qilin Wang, Zhengkai Jiang, Chengming Xu, Jiangning Zhang, Yabiao Wang, Xinyi Zhang, Yun Cao, Weijian Cao, Chengjie Wang, and Yanwei Fu. VividPose: Advancing stable video diffusion for realistic human image animation. In *ICME*, 2024b.

Qixun Wang, Xu Bai, Haofan Wang, Zekui Qin, and Anthony Chen. InstantID: Zero-shot identity-preserving generation in seconds. *arXiv preprint arXiv:2401.07519*, 2024c.

Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. Video-to-video synthesis. In *NeurIPS*, 2018.

Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *NeurIPS*, 2019.

Xiang Wang, Shiwei Zhang, Changxin Gao, Jiayu Wang, Xiaoqiang Zhou, Yingya Zhang, Luxin Yan, and Nong Sang. UniAnimate: Taming unified video diffusion models for consistent human image animation. *Science China Information Sciences*, 2025a.

Xiang Wang, Shiwei Zhang, Longxiang Tang, Yingya Zhang, Changxin Gao, Yuehuan Wang, and Nong Sang. UniAnimate-DiT: Human image animation with large-scale video diffusion transformer. arXiv preprint arXiv:2504.11289, 2025b.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.

Yujie Wei, Shiwei Zhang, Zhiwu Qing, Hangjie Yuan, Zhiheng Liu, Yu Liu, Yingya Zhang, Jingren Zhou, and Hongming Shan. DreamVideo: Composing your dream videos with customized subject and motion. In *CVPR*, 2024.

Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, 2022.

Zhen Xing, Qi Dai, Han Hu, Zuxuan Wu, and Yu-Gang Jiang. SimDA: Simple diffusion adapter for efficient video generation. In *CVPR*, 2024.

Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. MagicAnimate: Temporally consistent human image animation using diffusion model. In *CVPR*, 2024.

Haiwei Xue, Xiangyang Luo, Zhanghao Hu, Xin Zhang, Xunzhi Xiang, Yuqin Dai, Jianzhuang Liu, Zhensong Zhang, Minglei Li, Jian Yang, Fei Ma, Zhiyong Wu, Changpeng Yang, Zonghong Dai, and Fei Richard Yu. Human motion video generation: A survey. In *TPAMI*, 2025.

Yuxuan Yan, Chi Zhang, Rui Wang, Yichao Zhou, Gege Zhang, Pei Cheng, Gang Yu, and Bin Fu. FaceStudio: Put your face everywhere in seconds. *arXiv preprint arXiv:2312.02663*, 2023.

Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *ICCV Workshops*, 2023.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

Hu Ye, Jun Zhang, Siyi Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Polina Zablotskaia et al. DwNet: Dense warp-based network for pose-guided human video generation. In *BMVC*, 2019.

Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *ICCV*, 2019.

Yu Zeng, Vishal M Patel, Haochen Wang, Xun Huang, Ting-Chun Wang, Ming-Yu Liu, and Yogesh Balaji. JeDi: Joint-image diffusion models for finetuning-free personalized text-to-image generation. In *CVPR*, 2024.

Bowen Zhang, Chenyang Qi, Pan Zhang, Bo Zhang, HsiangTao Wu, Dong Chen, Qifeng Chen, Yong Wang, and Fang Wen. MetaPortrait: Identity-preserving talking head generation with fast personalized adaptation. In *CVPR*, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Xuanmeng Zhang, Jianfeng Zhang, Huichao Zhang, Jun Hao Liew, Chenxu Zhang, Yi Yang, and Jiashi Feng. AvatarStudio: High-fidelity and animatable 3D avatar creation from text. *IJCV*, 2025a.

Yuang Zhang, Jiaxi Gu, Li-Wen Wang, Han Wang, Junqi Cheng, Yuefeng Zhu, and Fangyuan Zou. MimicMotion: High-quality human motion video generation with confidence-aware pose guidance. In *ICML*, 2025b.

Haoyu Zhao, Zhongang Qi, Cong Wang, Qingping Zheng, Guansong Lu, Fei Chen, Hang Xu, and Zuxuan Wu. DynamiCtrl: Rethinking the basic structure and the role of text for high-quality human image animation. *arXiv:2503.21246*, 2025.

Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jia-Wei Liu, Weijia Wu, J. Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. In *ECCV*, 2024.

Jingkai Zhou, Yifan Wu, Shikai Li, Min Wei, Chao Fan, Weihua Chen, Wei Jiang, and Fan Wang. RealisDance-DiT: Simple yet strong baseline towards controllable character animation in the wild. *arXiv preprint arXiv:2504.14977*, 2025.

APPENDIX

TABLE OF CONTENTS

A Further Implementation Details 17
A.1 Base Model Architectures 17
A.2 Efficient Adaptation via LoRA 17
A.3 Training Details 19

B Evaluation Metrics 20

C Further Ablation 21
C.1 Additional Quantitative Results 21
C.2 Ablation on Secondary Model 22
C.3 Cross-Driving Evaluation 23

D Dataset Curation & Evaluation Protocol 24
D.1 Unseen100-Val Dataset 24
D.2 Evaluation Protocol 25

E Additional Qualitative Results 26

F User Preference Study 29

G Limitations and Failure Cases 31

A Further Implementation Details

A.1 Base Model Architectures

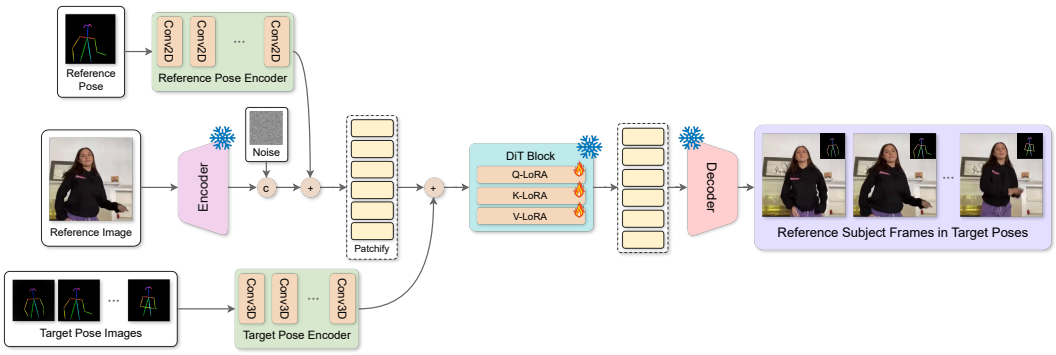

Figure 5: **Overview of the base model architecture.** Built on the Wan2.1-14B DiT backbone, we adapt from our base model by freezing the VAE, Reference Image Encoder, and main DiT parameters to ensure efficient adaptation. Conditioning utilizes a 3D convolutional encoder for driving poses and a 2D encoder for reference alignment. The pretrained LoRA modules in the DiT attention and feed-forward layers (the figure depicts the attention projections) are adapted only at the selected components.

To demonstrate the model-agnostic nature of AdaptAnyone, we apply our framework to two architecturally distinct pose-guided video diffusion backbones.

UniAnimate-DiT (Wang et al., 2025b) is built on the Wan2.1-14B video diffusion transformer (Wan-DiT) (Wan et al., 2025). As illustrated in Figure 5, the model operates in a compressed latent space provided by a Video VAE. Conditioning is supplied through three streams: (i) a *reference appearance* stream, where the source image is encoded via the frozen VAE encoder and concatenated with the noisy video latents to preserve high-frequency appearance details; (ii) a *target pose conditioner*, consisting of stacked 3D convolutional layers that capture the temporal dynamics and spatial structure of the pose driving skeleton; and (iii) a *reference pose conditioner*, a 2D encoder that processes the pose skeleton of the reference image to provide alignment information between the static appearance and the initial generated frame. The DiT backbone contains 40 transformer blocks with full-attention layers.

DynamiCtrl (Zhao et al., 2025) is built on the CogVideoX-5B backbone (Yang et al., 2025). Unlike UniAnimate-DiT, DynamiCtrl does not employ a dedicated expert pose encoder. Instead, it reuses the pre-trained VAE encoder to encode both the reference image and the driving pose sequence into a shared latent space, simplifying the architecture. Pose features are injected into the DiT via a Pose-adaptive Layer Norm (PadaLN), which learns per-layer scale, shift, and gate parameters from the diffusion timestep embedding and applies them to the normalized pose tokens before adding them to the visual token stream. DynamiCtrl additionally preserves text conditioning through a "Joint-text" paradigm, where textual tokens from a T5 encoder are fused with visual tokens via full-attention blocks. While DynamiCtrl was originally designed for per-video prompts generated by Qwen2-VL (Wang et al., 2024a), we use the same fixed prompt as for UniAnimate-DiT (Appendix D.2). The backbone contains 42 transformer blocks.

In both backbones, we freeze all pre-trained parameters (VAE, pose/text encoders, and DiT weights) and adapt only lightweight LoRA modules, as described next.

A.2 Efficient Adaptation via LoRA

For both base models, we adapt Low-Rank Adaptation (LoRA) (Hu et al., 2022) parameters only in DiT components selected by the sensitivity-guided placement strategy (Section 4.2), retaining the components with the highest combined sensitivity for each architecture. As illustrated in Figures 4 & 6, for UniAnimate-DiT, we retain 29 of its 120 components; for DynamiCtrl, we retain 9 of 42 blocks. For UniAnimate-DiT, θ is the pretrained UniAnimate-DiT LoRA, which is attached to every

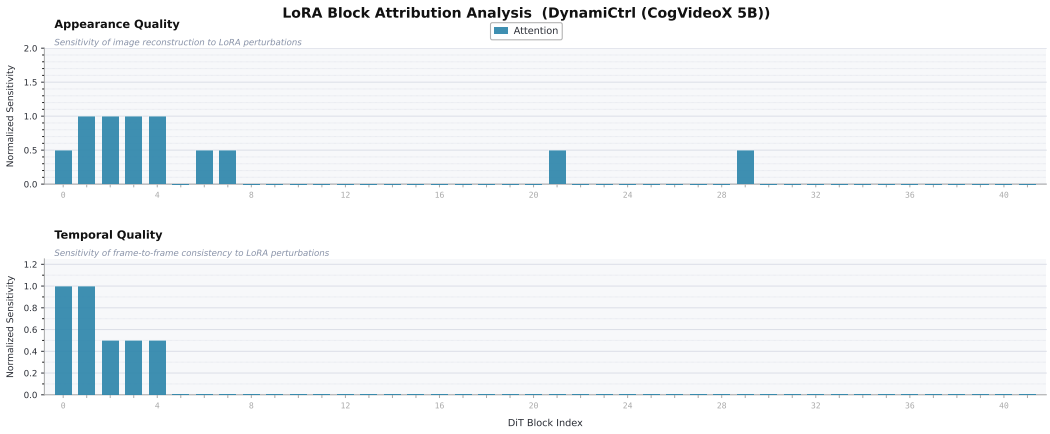

Figure 6: Attribution graph for the DynamiCtrl model. Based on our sensitivity-guided placement strategy, we inject fresh LoRA modules into only 9 selected blocks, ensuring highly efficient adaptation. Scores are finite differences of bfloat16 losses, normalized to their maximum; they are therefore quantized, and zero indicates a change below bfloat16 resolution.

DiT block; adaptation updates only the LoRA weights of the selected components, and all other LoRA weights keep their pretrained values.

Sensitivity Analysis. For each component, we add Gaussian noise with standard deviation $\epsilon{=}10^{-3}$ to its LoRA weights; since the pre-trained UniAnimate-DiT LoRA weights have a median magnitude of about $4\text{–}9\times10^{-3}$, this is roughly 11–25% of a typical weight. We measure the resulting change in two latent-space quantities, each divided by ϵ: *appearance* sensitivity is the change in the diffusion MSE between prediction and target over all frames, and *temporal* sensitivity is the change in the variance of the per-frame MSE plus the mean MSE between consecutive predicted frames. Scores are computed on TikTok training videos. Components are ranked by the unweighted sum of the two scores (normalizing each score by its maximum selects the same components), and the top 29 are adapted. Losses are evaluated in bfloat16, so scores are quantized, and components whose loss change falls below bfloat16 resolution score zero. The 29 adapted components are therefore the 27 with a non-zero appearance score and the two late-block components (block 38 FFN, block 39 self-attention) with the largest temporal scores (Figure 4).

DynamiCtrl thus requires fewer adaptation points than the Wan-based model. This is potentially due to the fact that Wan was already heavily fine-tuned with LoRA, whereas we inject fresh LoRA modules into DynamiCtrl here, allowing it to capture holistic appearance and motion dynamics with fewer injections.

By applying this sensitivity-guided selection, the total trainable LoRA parameters are approximately 37 M for UniAnimate-DiT and approximately 3 M for DynamiCtrl, representing less than 1% of each respective backbone.

The two backbones differ in their internal block structure, which affects where LoRA modules are placed:

UniAnimate-DiT. Each DiT block contains three component types: self-attention, cross-attention, and a feed-forward network (FFN). Its pretrained LoRA covers the query, key, value, and output projections of both attention mechanisms, as well as the input and output projections of the FFN; we adapt it only at the selected components.

DynamiCtrl. Each DiT block contains a single full-attention layer (which jointly handles spatial and temporal mixing) and a feed-forward network. Because the FFN layers in CogVideoX-5B are substantially larger than the attention projections, injecting LoRA into both components would introduce excessive trainable parameters that harm adaptation efficiency. We therefore inject LoRA only into the attention projections (query, key, value, and output) of each selected block, leaving the FFN layers frozen.

Rank & Scaling. The LoRA rank r and scaling factor (set equal to r, following standard convention) are set per-backbone: rank 32 for UniAnimate-DiT and rank 16 for DynamiCtrl (the effect of rank is analyzed in Tables 9 and 15).

Table 7 summarizes the parameter distribution for both backbones. After sensitivity-guided selection, the total trainable LoRA parameters are approximately 37 M for UniAnimate-DiT and approximately 3 M for DynamiCtrl, representing less than 1% of each respective backbone.

Table 7: Model parameter distribution for both base models. All components except the LoRA adapters are frozen during meta-personalization training and test-time adaptation.

Component Type	UniAnimate-DiT	DynamiCtrl	Mode
Denoising Backbone DiT	14.0 B	5.0 B	frozen
Video VAE Encoder + Decoder	0.13 B	0.4 B	frozen
Pose Encoder(s)	381 K	—	frozen
Text Encoder	5.7 B	4.8 B	frozen
Image Encoder (CLIP)	0.6 B	—	frozen
Subject Adapters (LoRA)	∼37 M	∼3.05 M	train
LoRA rank / scale	32 / 32	16 / 16	—
Selected units	29 / 120 components	9 / 42 blocks	—

A.3 Training Details

Training Resolution. Each base model is trained at its native resolution to preserve the pre-trained spatial priors. UniAnimate-DiT is trained at 832×480 and DynamiCtrl at 1360×768. As noted in Section D.2, all generated videos are center-cropped and resized to 512×512 for metric computation to ensure a fair comparison.

Optimizer & Learning Rates. Both backbones use Adam (Kingma & Ba, 2015) for the generalization loop and plain gradient descent for the subject adaptation loop. The generalization-loop (β) and subject adaptation-loop (α) learning rates are set to 1×10^{-5} and 1×10^{-4} for UniAnimate-DiT, and 1×10^{-5} and 3×10^{-4} for DynamiCtrl, respectively.

Meta-Personalization Schedule. Training runs for 10,000 generalization-loop steps with a batch size of 1, distributed across $4\times$ NVIDIA H100-80G GPUs. We employ gradient checkpointing with CPU offloading for UniAnimate-DiT to fit the 14B backbone within GPU memory. No learning rate warmup or decay is applied; no data augmentation (e.g., random cropping, color jitter, or horizontal flipping) is used, ensuring that the adaptation prior is learned entirely from the raw video frames.

Task Sampling. At each meta-step, a single training video is sampled as a task. Frames are split into support and query sets via uniform sampling: $\mathcal{M}$ frames are drawn at evenly spaced intervals for the support set $\mathcal{D}_{sup}$, and the remaining frames form the query set $\mathcal{D}_{qry}$.

Wall-Clock Time. On $4\times$ H100-80G GPUs, a single meta-step takes approximately 25 s for UniAnimate-DiT and 7.9 s for DynamiCtrl. The full 10,000-step meta-personalization training therefore completes in approximately 69 hours (∼2.9 days) for UniAnimate-DiT and 22 hours (∼0.9 days) for DynamiCtrl. This is a one-time cost; once the adaptation prior θ_0 is obtained, test-time personalization for any new subject requires only $\mathcal{K}$=3 gradient steps (3.6 s for UniAnimate-DiT and 7.8 s for DynamiCtrl on a single GPU).

Baselines. Naive TTA applies exactly our test-time procedure, adapting the same LoRA parameters with the same optimizer and learning rate, but starts from the pretrained weights instead of our learned initialization. The comparison therefore isolates what meta-learning contributes; Table 3 sweeps its budget from 3 to 1500 steps. LoRA SFT (Table 4) fine-tunes the pretrained UniAnimate-DiT LoRA on the same training data for the same number of steps with AdamW (learning rate 10^{-5}), followed by the same $\mathcal{K}$=3-step test-time adaptation.

Algorithm 2 AdaptAnyone test-time personalization (Inference)

Input: $\mathcal{M}$ reference frames V_{new} of new subject S_{new}, target poses c_{pose}
Pre-trained backbone: frozen video diffusion backbone Φ
Trainable parameters: LoRA adaptation prior θ_0
Hyperparameters: subject adaptation loop learning rate α; number of adaptation steps $\mathcal{K}$ (e.g., $\mathcal{K} = 3$)

$\theta_{new} \leftarrow \theta_0$ ▷ Start from adaptation prior
▷ **— Step 1: Adapt —**
for $k = 1$ to $\mathcal{K}$ **do**
  $\mathcal{L}_{adapt} \leftarrow \mathcal{L}_{simple}(v_{\Phi,\theta_{new}}, V_{new})$
  $\theta_{new} \leftarrow \theta_{new} - \alpha \nabla_{\theta_{new}} \mathcal{L}_{adapt}$
end for
▷ **— Step 2: Generate —**
$\hat{V} \leftarrow \text{Sample}(v_{\Phi,\theta_{new}}, c_{pose})$ ▷ Generate new video
return $\hat{V}$

B Evaluation Metrics

We provide detailed formulations of all evaluation metrics used in our experiments. Let I_t and $\hat{I}_t$ denote the ground-truth and generated frames at time step t, respectively, with spatial dimensions $H \times W$, C color channels, and T total frames.

PSNR (Hore & Ziou, 2010). We first compute the per-frame mean squared error:

$$\text{MSE} = \frac{1}{HWC} \sum_{x=1}^{W} \sum_{y=1}^{H} \sum_{c=1}^{C} \big(I_t(x,y,c) - \hat{I}_t(x,y,c)\big)^2, \tag{4}$$

and then obtain the peak signal-to-noise ratio as:

$$\text{PSNR} = 20 \log_{10} \left(\frac{255}{\sqrt{\text{MSE}}} \right). \tag{5}$$

Higher values indicate better reconstruction fidelity.

SSIM (Wang et al., 2004). For each frame t and color channel c, we compute the structural similarity:

$$\text{SSIM}_t^c = \text{SSIM}\big(I_t(\cdot,\cdot,c),\ \hat{I}_t(\cdot,\cdot,c)\big). \tag{6}$$

The reported score is the average over all frames and channels:

$$\text{SSIM} = \frac{1}{TC} \sum_{t=1}^{T} \sum_{c=1}^{C} \text{SSIM}_t^c. \tag{7}$$

LPIPS (Zhang et al., 2018). For each frame, let $\phi_\ell(\cdot)$ denote the feature map extracted from the ℓ-th layer of a pre-trained VGG backbone, and let w_ℓ denote the learned per-channel weights. The per-frame perceptual distance is:

$$l_t = \frac{1}{L} \sum_{\ell=1}^{L} \frac{1}{H_\ell W_\ell} \left\| w_\ell \odot \big(\phi_\ell(I_t) - \phi_\ell(\hat{I}_t)\big) \right\|_1, \tag{8}$$

where H_ℓ and W_ℓ are the spatial dimensions of the ℓ-th feature map. The video-level LPIPS is averaged across all frames:

$$\text{LPIPS} = \frac{1}{T} \sum_{t=1}^{T} l_t. \tag{9}$$

Lower values indicate greater perceptual similarity.

FID (Heusel et al., 2017). We evaluate frame-level distributional quality using activations from a pre-trained Inception-V3 network. Let $(\mu_r^{\text{fid}}, \Sigma_r^{\text{fid}})$ and $(\mu_f^{\text{fid}}, \Sigma_f^{\text{fid}})$ denote the Gaussian statistics fitted to the real and generated frame features, respectively:

$$\text{FID} = \left\| \mu_r^{\text{fid}} - \mu_f^{\text{fid}} \right\|_2^2 + \text{Tr}\Big(\Sigma_r^{\text{fid}} + \Sigma_f^{\text{fid}} - 2\big(\Sigma_r^{\text{fid}}\, \Sigma_f^{\text{fid}}\big)^{1/2} \Big). \tag{10}$$

FVD (Unterthiner et al., 2018). To assess temporal realism, we extract spatiotemporal features using an Inflated 3D ConvNet (I3D) (Carreira & Zisserman, 2017). Each video is divided into non-overlapping 16-frame segments. Let $(\mu_r^{\text{fvd}}, \Sigma_r^{\text{fvd}})$ and $(\mu_f^{\text{fvd}}, \Sigma_f^{\text{fvd}})$ denote the Gaussian statistics of real and generated segments:

$$\text{FVD} = \left\|\mu_r^{\text{fvd}} - \mu_f^{\text{fvd}}\right\|_2^2 + \text{Tr}\Big(\Sigma_r^{\text{fvd}} + \Sigma_f^{\text{fvd}} - 2\big(\Sigma_r^{\text{fvd}}\,\Sigma_f^{\text{fvd}}\big)^{1/2}\Big)\,. \tag{11}$$

FID-VID (Balaji et al., 2019). While FVD operates on fixed-length segments, FID-VID captures long-range temporal consistency by aggregating I3D clip-level embeddings across the entire video into a single descriptor per sequence. The Gaussian statistics (μ_r, Σ_r) and (μ_f, Σ_f) are then estimated over these video-level descriptors:

$$\text{FID-VID} = \|\mu_r - \mu_f\|_2^2 + \text{Tr}\Big(\Sigma_r + \Sigma_f - 2\big(\Sigma_r\,\Sigma_f\big)^{1/2}\Big)\,. \tag{12}$$

Lower values for FID, FVD, and FID-VID indicate closer distributional alignment with real videos.

Adaptation Time. The wall-clock time of the test-time gradient steps on a single GPU, reported in minutes. For DynamiCtrl, Naive TTA times at intermediate budgets are linearly interpolated from the measured 3- and 1500-step runs.

C Further Ablation

C.1 Additional Quantitative Results

Table 8: **Ablation on the number of support frames ($\mathcal{M}$).** Performance peaks at $\mathcal{M}$=5 across most metrics, confirming that our meta-learned initialization maximizes identity information from a minimal frame budget. Beyond $\mathcal{M}$=5, performance declines.

Method	$\mathcal{M}$	PSNR↑	SSIM↑	LPIPS↓	FID↓	FID-VID↓	FVD↓
Base Model	N/A	19.35	0.765	0.235	28.47	9.12	402.14
Ours	1	19.48	0.784	0.229	25.31	8.94	341.28
	3	19.97	**0.832**	0.217	19.07	7.89	178.86
	5	**20.36**	0.829	**0.201**	**18.42**	**7.29**	**177.51**
	7	20.21	0.826	0.205	19.28	7.68	183.42
	10	19.89	0.818	0.212	21.15	8.21	212.67

Number of Support Frames ($\mathcal{M}$). Table 8 varies the number of reference frames provided. Even a single frame ($\mathcal{M}$=1) improves over the base model, demonstrating that our meta-initialization extracts useful appearance signal from minimal data. Performance peaks at $\mathcal{M}$=5 and declines with more frames. This confirms our method is highly data-efficient. We use $\mathcal{M}$=3 as default to reflect a more practical deployment scenario.

Table 9: **Ablation on the LoRA rank.** The LoRA scaling factor equals the rank. ↑ indicates higher is better, and ↓ lower is better.

LoRA Rank	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	FID-VID↓
4	19.28	0.778	0.234	26.83	372.41	9.04
8	19.41	0.793	0.228	24.15	301.28	8.72
16	19.72	0.818	0.221	21.42	215.63	8.21
32	**19.97**	**0.832**	**0.217**	**19.07**	**178.86**	**7.89**
64	19.81	0.824	0.219	20.38	198.52	8.11
128	19.62	0.809	0.224	22.76	247.19	8.48

LoRA Rank. Table 9 examines the effect of LoRA rank (with the scaling factor equal to the rank). Performance follows an inverted-U pattern: low ranks (4, 8) lack the capacity to encode appearance in three gradient steps, while high ranks (64, 128) introduce excess parameters that overfit to frame-specific details. Rank 32 achieves the optimal balance, with degradation at rank 128 being particularly visible on FVD, consistent with temporal overfitting from an over-parameterized adaptation.

Table 10: **Ablation on learning rates.** We vary the subject adaptation rate (α) and generalization meta-learning rate (β) independently.

Subject α	Generalization β	PSNR↑	SSIM↑	LPIPS↓	FID↓	FID-VID↓	FVD↓
1e-3	1e-5	19.08	0.774	0.241	26.41	9.87	341.26
5e-4	1e-5	19.61	0.811	0.224	21.83	8.41	224.15
1e-4	**1e-5**	**19.97**	**0.832**	**0.217**	**19.07**	**7.89**	**178.86**
1e-4	5e-5	19.74	0.821	0.221	20.62	8.18	203.47
1e-4	1e-4	19.38	0.793	0.231	24.15	9.24	289.63

Learning Rates (α, β). Table 10 examines the effect of learning rates. The model is notably more sensitive to subject adaptation rate changes: increasing α by $10\times$ (to 1e-3) degrades FVD by $1.9\times$, whereas increasing β by the same factor causes a milder $1.6\times$ degradation. This asymmetry is expected, as α directly controls the per-subject adaptation dynamics while β only affects the offline meta-initialization. Overly aggressive α causes the 3-step adaptation to overshoot, while overly aggressive β produces a less stable initialization that inner adaptation cannot fully compensate for.

Seed Reproducibility. Table 11 repeats the TikTok evaluation of UniAnimate-DiT with three random seeds. AdaptAnyone has a lower standard deviation than the base model on five of six metrics, most notably on FVD (26.2 vs. 119.1) and FID-VID (0.49 vs. 4.34).

Table 11: **Seed reproducibility** (TikTok, UniAnimate-DiT). Reported results with the standard deviation over three random seeds.

Method	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	FID-VID↓
Base	19.35±0.37	0.765±0.018	0.235±0.026	28.47±1.19	402.14±119.1	9.12±4.34
AdaptAnyone	19.97±0.18	0.832±0.003	0.217±0.003	19.07±1.31	178.86±26.2	7.89±0.49

C.2 Ablation on secondary model

Table 12: **AdaptAnyone vs. Naive TTA at varying fine-tuning budgets for DynamiCtrl.**

Model	Fine-Tuning Steps	Fine-Tuning time (min)	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	FID-VID↓
Base	0	N/A	18.26	0.804	0.247	31.21	502.35	25.62
Naive TTA	3	0.13	17.43	0.806	0.264	43.17	441.28	31.47
	100	4.30	17.52	0.810	0.258	38.42	412.53	33.82
	200	8.60	17.71	0.813	0.255	36.89	398.17	34.15
	500	21.50	17.94	0.816	0.249	33.74	381.62	32.61
	1000	43.00	18.12	0.819	0.243	31.52	367.45	28.73
	1500	64.50	18.21	0.821	0.238	29.87	358.91	23.41
AdaptAnyone	**3**	**0.13**	**18.85**	**0.827**	**0.219**	**28.84**	**349.21**	**17.21**

Comparison against Naive TTA. Table 12 compares AdaptAnyone against Naive TTA on the DynamiCtrl backbone. Consistent with the UniAnimate-DiT results in Table 3, our 3-step adaptation surpasses even 1500-step Naive TTA across all metrics.

LoRA Placement Strategy. Table 13 validates the sensitivity-guided LoRA placement strategy on DynamiCtrl's 42-block architecture. As with UniAnimate-DiT (Table 6), adapting only the last half of blocks performs worse, while sensitivity-guided selection outperforms full injection across all metrics. Notably, our strategy selects only 9 of 42 blocks (see Figure 6), a sparser selection than on UniAnimate-DiT (29 of 120 components). This is because each block contains only a single attention layer (no separate cross-attention), sensitivity is concentrated in a smaller set of early blocks, with sparse contributions from mid-range and later blocks. Despite adapting fewer than a quarter of all blocks, the sensitivity-guided configuration achieves the best results across all metrics.

Table 13: **Ablation on LoRA placement strategy for DynamiCtrl.** We compare heuristic block selection strategies against our sensitivity-guided approach to identify the most impactful component types and blocks.

Strategy	# Blocks	PSNR↑	SSIM↑	LPIPS↓	FID↓	FID-VID↓	FVD↓
Base Model (no LoRA)	0	18.26	0.804	0.247	31.21	25.62	502.35
All Blocks	42	18.51	0.817	0.231	29.58	20.37	411.83
First 21	21	18.43	0.811	0.235	30.41	23.52	466.71
Last 21	21	18.17	0.806	0.248	31.03	25.18	492.16
Sensitivity-Guided	**9**	**18.85**	**0.827**	**0.219**	**28.84**	**17.21**	**349.21**

Table 14: **Ablation on the number of support frames ($\mathcal{M}$) for DynamiCtrl.**

Method	$\mathcal{M}$	PSNR↑	SSIM↑	LPIPS↓	FID↓	FID-VID↓	FVD↓
Base Model	N/A	18.26	0.804	0.247	31.21	25.62	502.35
Ours	1	18.39	0.811	0.237	30.52	24.18	458.73
	3	18.85	**0.827**	0.219	28.84	17.21	349.21
	5	**19.18**	0.824	**0.196**	**28.51**	**14.83**	**345.87**
	7	19.04	0.822	0.202	29.03	16.14	355.18
	10	18.74	0.819	0.213	29.52	19.03	375.41

Table 14 ablates the number of support frames $\mathcal{M}$. Performance peaks at $\mathcal{M}$=5, consistent with UniAnimate-DiT (Table 8). Also consistent is the SSIM behavior: SSIM peaks at $\mathcal{M}$=3 rather than $\mathcal{M}$=5 on both backbones, as additional frames dilute the per-frame structural alignment within the fixed 3-step budget. The degradation beyond $\mathcal{M}$=5 is more gradual than on UniAnimate-DiT, likely because the smaller backbone saturates earlier and thus has less capacity to overfit to excess reference frames. We use $\mathcal{M}$=3 as default to reflect a practical deployment scenario.

Table 15: **Ablation on the LoRA rank for DynamiCtrl.** The LoRA scaling factor equals the rank.

LoRA Rank	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	FID-VID↓
4	18.34	0.809	0.243	30.74	478.53	24.17
8	18.53	0.816	0.234	29.81	412.67	21.43
16	**18.85**	**0.827**	**0.219**	**28.84**	**349.21**	**17.21**
32	18.71	0.821	0.224	29.37	371.48	18.94
64	18.48	0.812	0.232	30.52	408.31	21.87
128	18.29	0.805	0.241	31.08	452.73	23.61

Table 15 ablates LoRA rank on DynamiCtrl. Unlike UniAnimate-DiT, which peaks at rank 32, DynamiCtrl achieves optimal performance at rank 16.

Table 16 ablates inner (α) and outer (β) learning rates on DynamiCtrl. The optimal inner rate shifts upward to 3e-4, compared to 1e-4 for UniAnimate-DiT (Table 10). This is attributable to the lower LoRA rank (16 vs. 32) and smaller backbone: fewer trainable parameters require larger per-parameter updates to traverse the same functional distance in three steps. The optimum is also flatter, with α=1e-3 performing nearly as well, suggesting the meta-initialization's loss landscape is smoother for the smaller model. The outer rate β follows the same pattern as UniAnimate-DiT: conservative values (1e-5) are preferred, as this is a property of FOMAML stability rather than backbone scale.

C.3 Cross-Driving Evaluation

Cross-Driving. Our main benchmarks evaluate self-driving, where the reference subject also performs the driving motion. To test whether the adapted model preserves the subject's identity when the motion comes from a different person, we follow the cross-reenactment protocol of X-UniMotion (Song et al., 2025) on 20 cross-pairs from Unseen100-Val, and measure identity (FaceID), body appearance (Body-Sim), and motion accuracy (Pose-Acc). As shown in Table 17, AdaptAnyone improves all three metrics over both the base model and 1500-step Naive TTA.

Table 16: **Ablation on learning rates for DynamiCtrl.**

Inner α	Outer β	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	FID-VID↓
1e-2	1e-5	17.89	0.791	0.253	33.47	468.21	26.83
1e-3	1e-5	18.78	0.824	0.221	28.92	354.37	17.58
3e-4	**1e-5**	**18.85**	**0.827**	**0.219**	**28.84**	**349.21**	**17.21**
1e-4	1e-5	18.52	0.814	0.231	29.71	397.83	20.42
5e-5	1e-5	18.31	0.808	0.239	30.48	451.29	23.17
3e-4	5e-5	18.69	0.820	0.225	29.23	368.74	18.53
3e-4	1e-4	18.41	0.809	0.235	30.14	421.38	22.71

Table 17: **Cross-driving identity and motion accuracy on Unseen100-Val**, following the cross-reenactment protocol of X-UniMotion (Song et al., 2025) (20 cross-pairs, reference $\neq$ driving subject). FaceID is ArcFace (Deng et al., 2019) cosine similarity. Body-Sim is DINOv2 (Oquab et al., 2024) image-feature cosine similarity and Pose-Acc is DWPose-PCK (Yang et al., 2023) with per-skeleton bone-length normalization (generated vs. driving keypoints), both being automated analogs of X-UniMotion's user-study FullID-Sim and Mot-Acc.

Method	FaceID↑	Body-Sim↑	Pose-Acc↑
Base (UniAnimate-DiT)	0.433	0.728	0.442
Naive TTA (1500 steps)	0.441	0.734	0.443
Ours	**0.459**	**0.754**	**0.456**

D Dataset Curation & Evaluation Protocol

D.1 Unseen100-Val Dataset

Motivation. While TikTok (Jafarian & Park, 2021) and UBC Fashion (Zablotskaia et al., 2019) are standard benchmarks for pose-guided human animation, both have significant limitations for evaluating modern video diffusion models. The TikTok test set contains only 10 video samples, which limits the diversity it can cover. UBC Fashion, though larger (100 test videos), is composed of simple catwalk sequences with uniform lighting and minimal background variation; as evidenced by the uniformly high scores across all methods in Table 1, this benchmark no longer discriminates effectively among current approaches. Neither dataset adequately covers the diversity of real-world deployment scenarios, including varying skin tones, challenging lighting conditions, and non-photorealistic subjects. To address these gaps, we curate **Unseen100-Val**, a 100-video zero-shot evaluation set designed to (a) better represent the visual diversity of real-world data and (b) stress-test models in challenging settings they have not been trained on, such as difficult lighting, complex backgrounds, and animated characters.

Collection & Eligibility. Videos were sourced from a mixture of TikTok and stock footage. Eligibility criteria required each video to feature a single person or character performing a dancing motion, with no heavy occlusion of any body part (slight tolerance is allowed for an arm briefly disappearing during fast motion). We additionally required that the DWPose (Yang et al., 2023) skeleton remain fully visible throughout the clip, ensuring reliable pose extraction for both adaptation and evaluation.

Composition & Diversity. Table 18 summarizes the dataset composition across several axes. Each attribute group independently covers all 100 videos; since categories overlap (e.g., a video can be both indoor and fast-motion), rows within each group sum to 100. We deliberately balance real and animated subjects and ensure broad coverage of appearance, motion complexity, and environmental conditions. Among the 80 real-human videos, we include a wide range of skin tones spanning East Asian, South Asian, Caucasian, and darker-skinned appearances to promote evaluation fairness. The 20 animated videos are evenly split between 2D (anime, cartoon) and 3D (rendered, stylized) characters, testing the model's ability to generalize beyond photorealistic subjects.

Preprocessing. All videos are center-cropped and resized to 1024×1024 resolution. Driving poses are extracted using DWPose (Yang et al., 2023), consistent with our training pipeline. The average clip duration is 14.4 seconds ($\pm$3.2 s), with a minimum of 8 s and a maximum of 24 s.

Table 18: Composition of the Unseen100-Val dataset. Each attribute group independently covers all 100 videos; rows within each group sum to 100 (or to 80 for appearance and setting, which applies only to real-human videos).

Attribute	Category	Count
Subject Type	Real	80
	Animated – 2D	10
	Animated – 3D	10
Setting	Indoor	54
	Outdoor	26
Appearance	East Asian	32
	South Asian	06
	Caucasian	24
	Darker-skinned	18
Motion Intensity	Slow / Moderate	48
	Fast / Complex	52
Lighting	Standard	61
	Challenging	39
Camera	Static	71
	Slight-Shift	29

Size Justification. At 100 videos, Unseen100-Val matches the test-set size of UBC Fashion while being 10× larger than TikTok's test split. Because the set is designed for zero-shot cross-domain evaluation rather than statistical power over narrow domains, 100 diverse videos provide sufficient coverage to surface meaningful failure modes and performance differences across methods.

Relation to the TikTok Dataset. Although part of Unseen100-Val is sourced from the TikTok platform, it shares no videos with the TikTok dataset (Jafarian & Park, 2021) used for training, and it deliberately covers conditions that the training data lacks, including outdoor scenes, challenging lighting, animated 2D and 3D characters, and diverse skin tones (Table 18).

D.2 Evaluation Protocol

Inference Settings. We follow the evaluation settings of RealisDance-DiT (Zhou et al., 2025), including its seed protocol. UniAnimate-DiT uses its default flow-matching sampler and DynamiCtrl uses DDIM (Song et al., 2021), each with 50 inference steps. The text prompt is fixed to "a person is dancing" for all experiments.

Resolution. While each base model is trained at its native resolution (see Section A), all generated videos are center-cropped and resized to 512×512 for metric computation to ensure a fair comparison across methods.

Support Frame Selection. Support frames are selected automatically; none are picked by hand. Within the support chunk, we run greedy farthest-point sampling in 2D keypoint space: starting from the reference frame (also the reference image used for generation), each step adds the frame whose minimum keypoint distance to the already-selected frames is largest, until $\mathcal{M}$ frames are chosen. The distance is the mean per-joint Euclidean distance between normalized DWPose whole-body keypoints (Yang et al., 2023). This spreads the support frames over the subject's range of poses and viewpoints. Each test video is split into two disjoint temporal chunks; support frames are selected from one chunk, and all metrics are computed only on the other, so support frames are never evaluated.

Baseline Details. For competing methods in Table 1, we report the numbers published in their papers (– where not reported). On Unseen100-Val (Table 2), we run all methods ourselves on the same clips with the same evaluation code. For DynamiCtrl, the original checkpoint was not publicly released despite being claimed in the paper; we therefore re-train the model using the official codebase, and all DynamiCtrl numbers reported in our tables reflect this re-trained checkpoint.

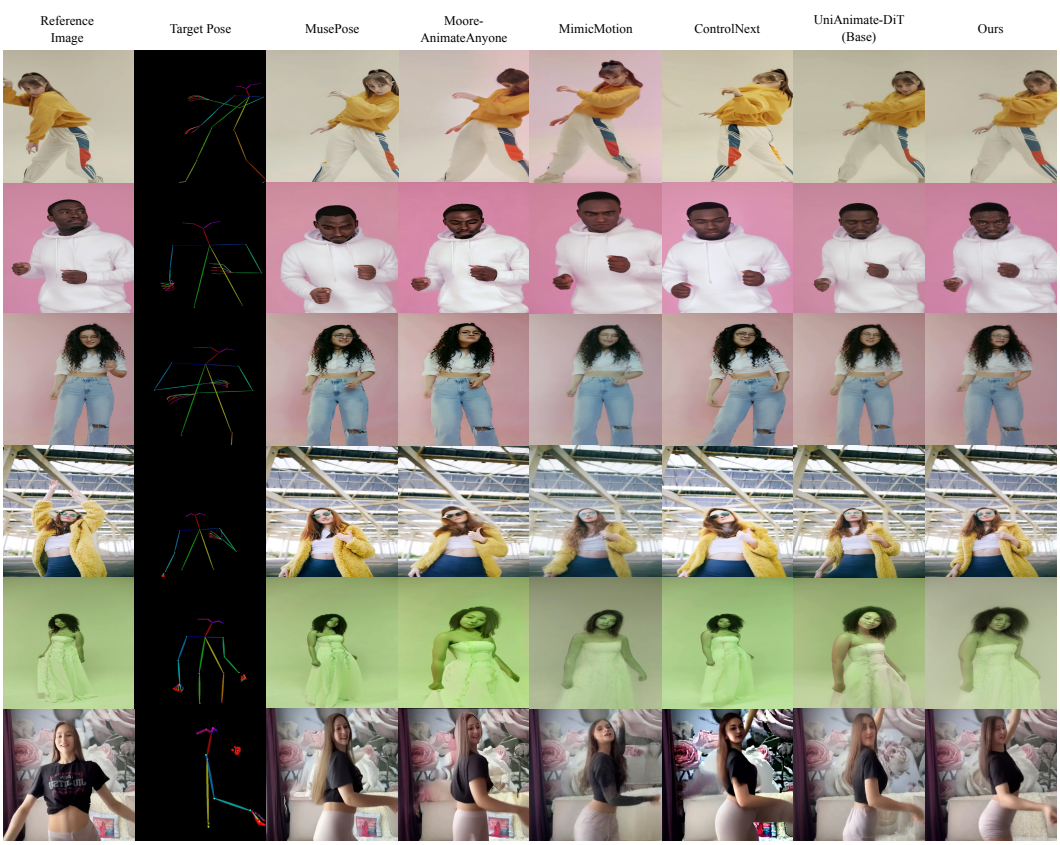

Figure 7: **In-domain comparisons: diverse motions and backgrounds.** Examples set illustrating variations in background setting, motion complexity, and clothing style. AdaptAnyone consistently preserves holistic appearance and pose alignment while producing sharper, more realistic details than competing methods.

E Additional Qualitative Results

In this section, we provide extended visual comparisons to further illustrate the effectiveness of AdaptAnyone across diverse subjects, motion types, and visual domains. Figures 7 and 8 present visual results covering a wide range of backgrounds, motion complexity, skin tones, lighting conditions, and clothing styles. Figure 9 extends the comparison to cross-domain animated characters from the Unseen100-Val evaluation set, including both 2D and 3D styles unseen during training. Across all examples, our 3-step adaptation consistently preserves the subject's holistic appearance while maintaining temporal coherence, even under challenging lighting conditions and complex motion sequences.

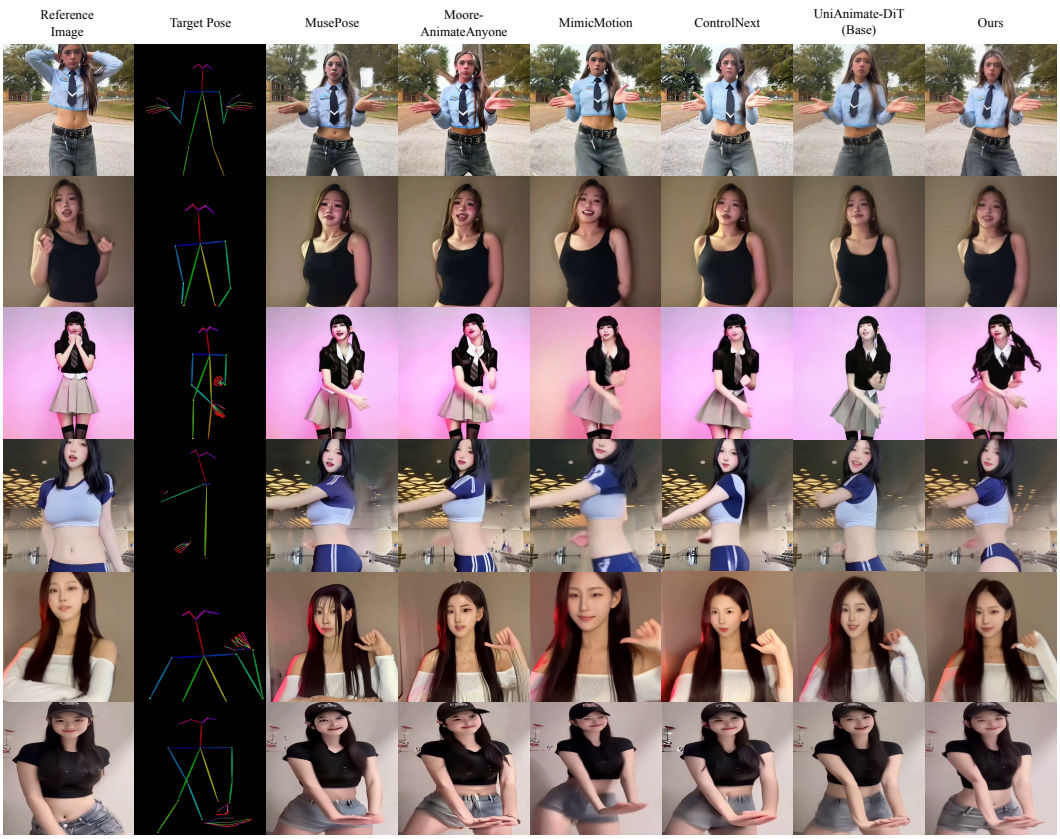

Figure 8: **In-domain comparisons: diverse appearance and illumination.** Examples covering variations in skin tone, lighting conditions, and clothing texture. AdaptAnyone maintains consistent subject appearance and realistic fine-grained details across all scenarios, clearly outperforming state-of-the-art methods.

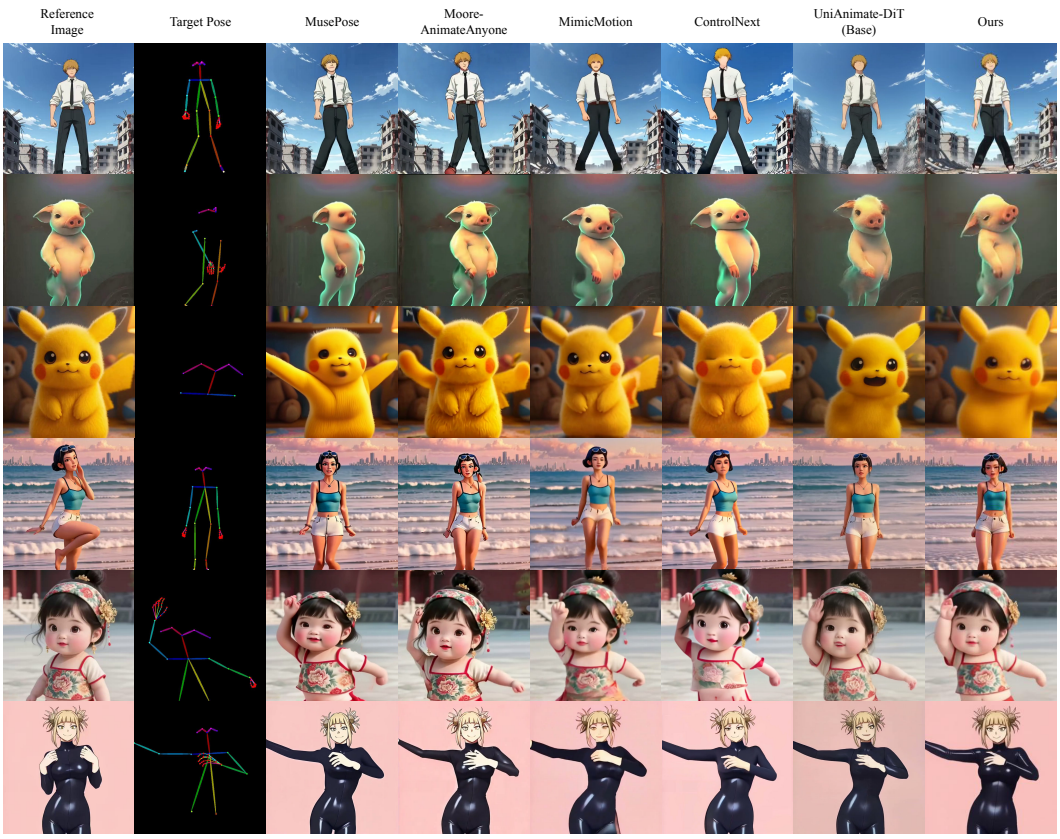

Figure 9: **Cross-domain comparisons: animated characters.** Examples from the Unseen100-Val evaluation set featuring 2D and 3D animated characters unseen during training. AdaptAnyone generalizes beyond photorealistic subjects, faithfully preserving stylized appearance details and character proportions while maintaining temporal coherence under complex motion.

F User Preference Study

To validate the perceptual quality of our results beyond numerical metrics, we conducted a subjective user study. We collected responses from 28 participants, consisting of both academic researchers in computer vision and general non-expert users, to ensure a balanced evaluation perspective.

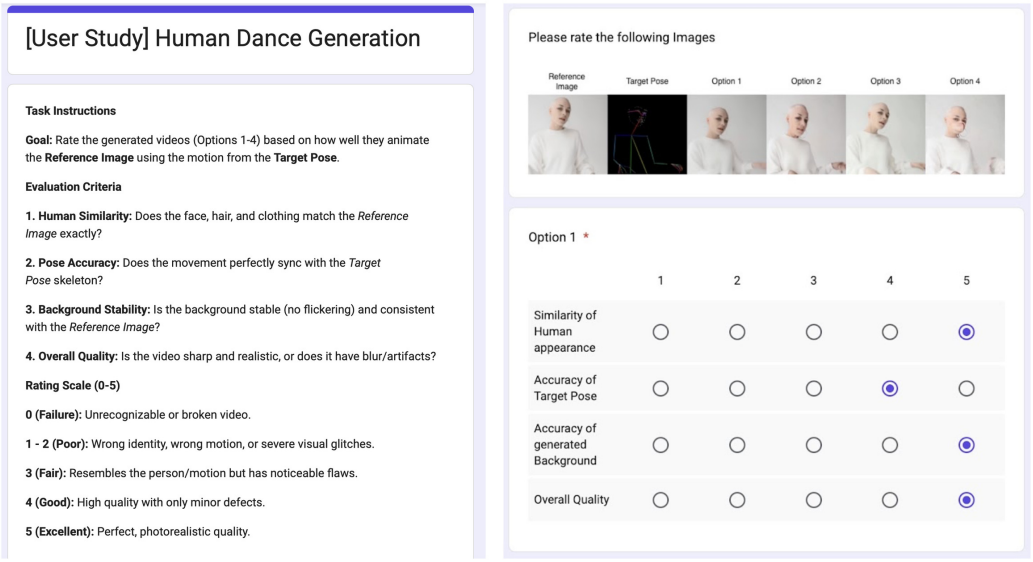

Figure 10: **Survey interface.** The interface of our user study. We provide detailed instructions, rating examples, and rating instances for users.

Protocol. The survey consisted of 15 video samples. For each, they were shown the Reference Image, the Target Pose sequence, and the anonymized generated outputs from four methods: our AdaptAnyone (UniAnimate-DiT version), the base UniAnimate-DiT model, Animate-X, and Moore-AnimateAnyone. As seen in Figure 10, participants were asked to rate each video on a scale from 1 (Worst) to 5 (Best) (the 0 level listed in the instructions was not selectable in the form) across four distinct categories:

- **Human Appearance Alignment:** Fidelity to the reference appearance.
- **Target Pose Alignment:** Synchronization and accuracy of the generated pose in reference to the target pose.
- **Background Alignment:** Stability and consistency of the background scene.
- **Overall Quality:** General perception.

Results Analysis. The aggregated response distributions are visualized in Figure 11. The curves demonstrate a clear trend in favor of our approach. Moore-AnimateAnyone and Animate-X are skewed towards the lower end of the scale (ratings 1–3), particularly for human appearance alignment, reflecting their difficulty in preserving an unseen subject's appearance. The base model shifts the distribution towards the center (ratings 3–4) but rarely reaches the top rating. Conversely, AdaptAnyone consistently dominates the high-score region. For all four categories, the probability density for our method peaks sharply at ratings of 4 or 5. This decisive skew indicates that human evaluators consistently perceive AdaptAnyone as offering superior appearance preservation, motion accuracy, and overall video quality compared to the baselines.

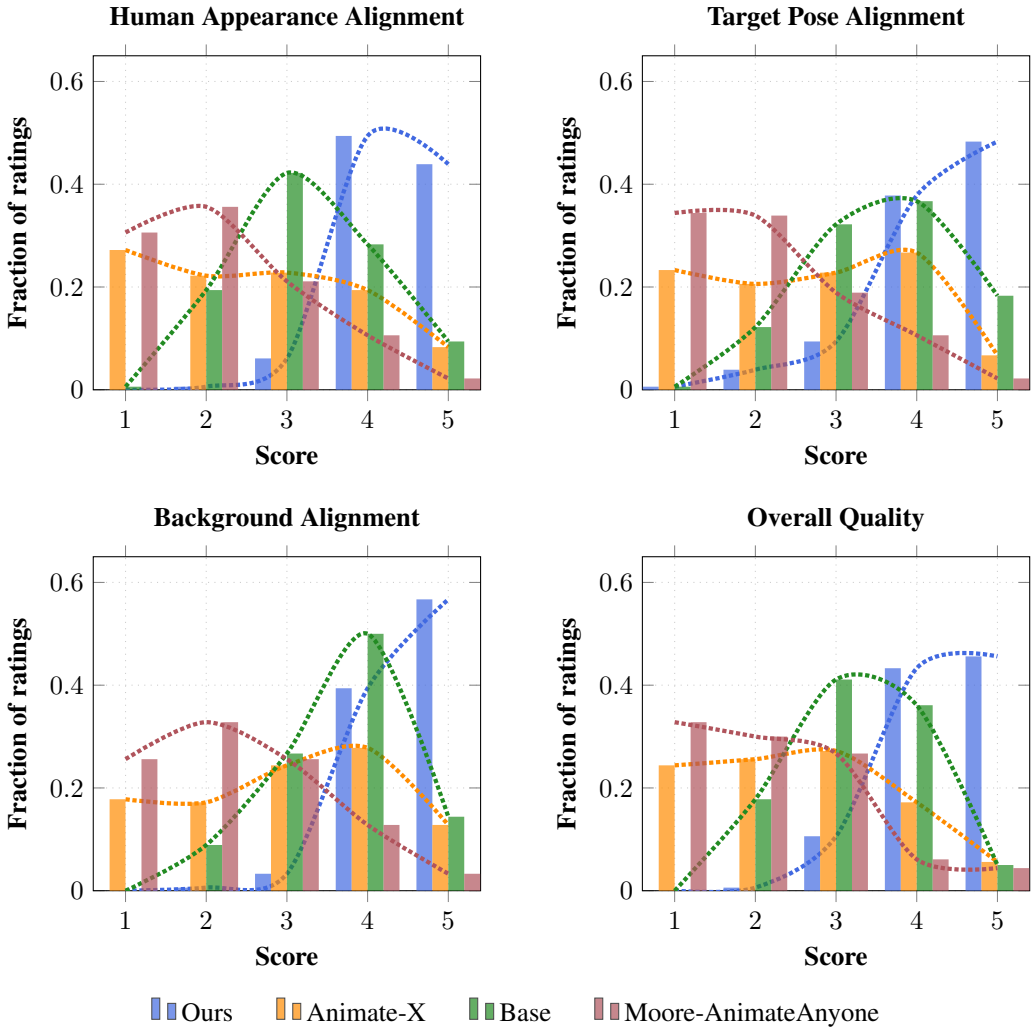

Figure 11: **Distribution of survey scores across categories.** Comparisons of Ours, Animate-X, Base, and Moore-AnimateAnyone methods (28 participants, 15 clips).

G LIMITATIONS AND FAILURE CASES

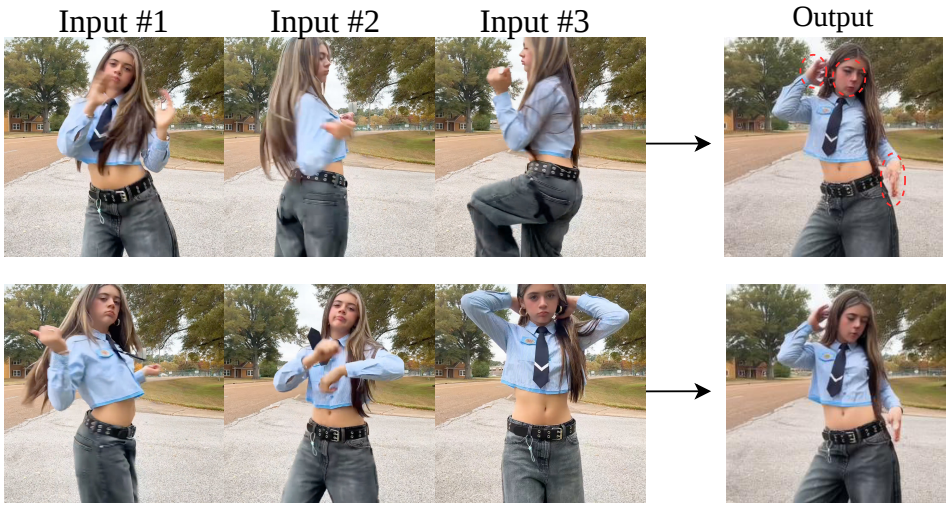

Figure 12: **Impact of support set quality.** We analyze the model's robustness to input imperfections. (Top) When the reference images contain both motion blur and occlusion, the adaptation process is compromised, resulting in a generated frame with washed-out details and lower appearance fidelity. (Bottom) In contrast, providing clear, high-quality reference frames allows AdaptAnyone to synthesize sharp, photorealistic results.

As discussed in Section 6, the performance of our test-time adaptation is highly dependent on the quality of the provided support frames. AdaptAnyone operates on the principle of extracting appearance cues from the few-shot examples; therefore, compromised inputs lead to compromised outputs. Figure 12 provides a direct visual comparison demonstrating this sensitivity: *Degraded Inputs (Top Row):* In this scenario, the reference images suffer from a combination of motion blur and occlusion (facial features are partially hidden or washed out). Consequently, the personalized model fails to lock onto sharp appearance features. The resulting frame is noticeably degraded, exhibiting a loss of facial texture and lower fidelity. *Clean Inputs (Bottom Row):* In contrast, when the support set consists of clear, unobstructed frames of the subject, the model successfully reconstructs high-frequency details and preserves appearance accurately.

This reinforces the need for the future work mentioned in our conclusion. Specifically, an adaptive frame selection module could filter out blurred or occluded samples before adaptation.