Data-Efficient Training and Adaptation of Target-Image-Aware Video Editing Models

Adeel Khan[1] **and Sajid Rafiq**[2]

[1]Department of Computer Science, University of Gujrat, Jalalpur Jattan Road, Gujrat 50700, Pakistan

[2]Department of Information Technology, The University of Haripur, Hattar Road, Haripur 22620, Pakistan

ABSTRACT

Video editing models that can adapt the content of a source video to match the identity, appearance, or style specified by a target image are increasingly important for controllable content creation, personalization, and post-production workflows. However, such target-image-aware video editing models typically depend on large collections of paired data or extensive per-target fine-tuning, which limits their practicality in scenarios where only a few reference images and possibly a short video are available. This paper examines data-efficient training and adaptation strategies for target-image-aware video editing models that leverage strong pretrained video generators and parameter-efficient conditioning mechanisms. The study formulates target-image-aware editing as a conditional transformation problem, where a pretrained backbone is adapted using small sets of target images and limited video samples while preserving temporal consistency and source motion. Several loss components are combined to balance fidelity to the target appearance, adherence to the source motion, and robustness to distribution shift between training and deployment domains. The paper investigates low-rank adaptation, hypernetwork-based modulation, and regularization by distillation from the base generator as mechanisms for reducing data requirements while maintaining editing quality. A theoretical analysis considers generalization from few examples and the interaction between adapter capacity and sample complexity. Empirical investigations on generic video editing scenarios illustrate how different components affect identity preservation, temporal coherence, and robustness to occlusions and large motions. The discussion highlights practical trade-offs between adaptation speed, parameter count, and performance under strict data budgets, and outlines directions for further improving data efficiency in video editing systems.

KEYWORDS

1. Introduction

Target-image-aware video editing aims to transform a source video so that foreground entities, global appearance, or stylistic attributes match those encoded in a reference image, while preserving the underlying motion and structural layout of the original sequence [1]. Typical instances of this task include identity-preserving face replacement driven by a portrait photograph, personalization of a character in animation using a concept art image, and restyling of a video according to a single frame depicting the desired aesthetics. These tasks require jointly modeling spatial fidelity to the target image and temporal coherence across frames in the edited video, under substantial variability in motion, lighting, occlusion, and viewpoint changes.

Recent progress in generative models for images and videos has provided strong backbones that can synthesize high-quality sequences conditioned on textual prompts, layout, or visual cues. However, directly training target-image-aware video editors on large-scale paired datasets is often impractical. Obtaining extensive collections of videos and corresponding target images

Article info:

Adeel Khan and Sajid Rafiq. *Data-Efficient Training and Adaptation of Target-Image-Aware Video Editing Models*. Nextqueries. Publishing

Table 1 Core variables in the target-image-aware video editing formulation.

Quantity	Symbol	Description
Source video	$V \in \mathbb{R}^{T\times H\times W\times C}$	Input sequence whose motion and spatial layout are to be preserved.
Target image	$I \in \mathbb{R}^{H_I\times W_I\times C}$	Reference appearance or style that should drive the edit.
Edited video	$\tilde{V} = \mathcal{F}_\theta(V, I)$	Output video that combines source motion with target appearance.
Ideal edited video	$V^\star$	Ground-truth or proxy supervision used during training.
Latent trajectory	$z = \phi(V) \in \mathbb{R}^{T\times D}$	Latent representation of the source video used by the generator.
Target embedding	$e = \psi(I) \in \mathbb{R}^{E}$	Compact representation of the target image fed to adapters or hypernetworks.

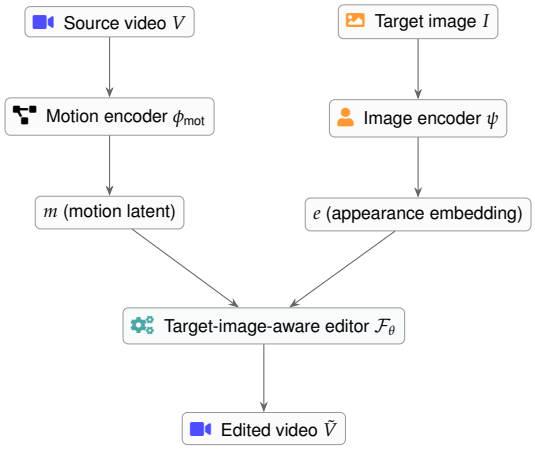

Figure 1 High-level formulation of target-image-aware video editing. A motion encoder extracts a motion latent m from the source video V, an image encoder extracts an appearance embedding e from the target image I, and a parameter-efficient editor $\mathcal{F}_\theta$ combines both signals to produce an edited video $\tilde{V}$ that follows the source motion while matching the target appearance.

that specify desired edits is expensive, and privacy or copyright considerations further limit the availability of data. In many applications, only a few images of the target identity or style are available, and only a limited set of videos can be used during adaptation. This setting raises a central question: how can one design video editing models and training procedures such that they are data-efficient, both in pretraining and in downstream adaptation to new targets?

In this work, data efficiency is considered under two complementary perspectives [2]. The first is global data efficiency, which concerns how much task-specific paired supervision is required to train a generic target-image-aware video editing model that can generalize to unseen source videos and novel target images. The second is per-target adaptation efficiency, which concerns how many examples are required to personalize the model to a new target identity, appearance, or style, given a pretrained backbone. Addressing both aspects is crucial: even if a powerful base model is available, per-target adaptation should rely on few reference images and little or no additional video footage, while maintaining control and consistency.

A key challenge stems from temporal coherence. Video editing models that operate frame by frame, without explicit temporal modeling, often produce flickering artifacts or inconsistent rendering of the target appearance across frames, especially when the target and source differ substantially in pose, expression, or illumination. At the same time, exploiting temporal information can easily lead to overfitting to the few observed sequences if the model has many degrees of freedom with respect to temporal dynamics. Data-efficient training therefore needs to regulate model capacity and exploit the temporal structure of motion in a way that stabilizes editing, without requiring large amounts of supervision.

Another challenge is disentangling motion, content, and appearance. In target-image-aware editing, the goal is to retain the motion trajectory and scene geometry of the source video while substituting or adapting attributes such as identity, tex-

Table 2 Encoders and latent spaces used for motion, appearance, and conditioning.

Component	Mapping	Latent space	Role
Video encoder	$\phi : V \mapsto z$	$\mathbb{R}^{T\times D}$	Maps source video to a latent trajectory for the generator.
Target encoder	$\psi : I \mapsto e$	$\mathbb{R}^{E}$	Extracts target appearance or style features.
Motion encoder	$\phi_{\text{mot}} : V \mapsto m$	$\mathbb{R}^{T\times D_m}$	Provides explicit motion features used for conditioning.
Identity/style head	$\eta_{\text{img}}(\cdot)$	Feature space	Defines identity or style consistency distances.
Motion descriptor	$\eta_{\text{mot}}(\cdot)$	Sequence space	Encodes optical flow, pose, or motion features for comparison.

ture, or style [3]. Pretrained video generators often entangle these factors in their latent representations, so naive conditioning on a target image can distort motion or alter background content. This work discusses modeling choices that separate motion latents from appearance latents and treat target-image features as modulating only the appearance channel, constrained by regularization terms that maintain structural consistency.

To address these issues, this paper proposes a formulation of target-image-aware video editing in which a pretrained video generator is augmented with parameter-efficient adapters that ingest target-image encodings and modulate selected layers of the backbone. Adaptation to a new target is performed by optimizing a small set of adapter parameters or by predicting adapter weights from the target image via a hypernetwork. The training objectives combine reconstruction-based losses, identity or style consistency losses, temporal smoothness penalties, and distillation terms that encourage the edited video to remain close to the distribution of outputs produced by the base generator when the edit is mild. These mechanisms aim to reduce the number of video-target pairs needed to train a generalizable editor and to allow per-target adaptation from a few reference images [4].

Beyond architectural and objective design, this study examines theoretical aspects of data efficiency in target-image-aware video editing. By modeling the adapter parameters as a low-dimensional, regularized perturbation of a pretrained model, it is possible to connect the number of target-specific samples required for stable adaptation with norms and ranks of these parameters. Approximate generalization bounds are derived that highlight the trade-off between adapter capacity, regularization strength, and the variance of the resulting edits [5]. Although the analysis relies on simplifying assumptions about the underlying data distributions and model linearizations, it provides guidance for choosing adapter structures that remain expressive but data-efficient.

The empirical section discusses experiments on synthetic and realistic editing scenarios, focusing on data regimes where only a few paired examples per target are available. The results compare different adapter designs, measures of identity preservation, and temporal metrics, and they illustrate how distillation and temporal regularization contribute to improved consistency. While the experiments are not exhaustive, they demonstrate characteristic behaviors of the proposed framework and motivate further exploration of data-efficient strategies for controllable video editing.

2. Problem Formulation

Consider a video represented as a tensor in a pixel or latent space. Let the source video be denoted by

$$V \in \mathbb{R}^{T\times H\times W\times C}, \tag{1}$$

where T is the number of frames, H and W are spatial dimensions, and C is the number of channels. Let the target image be denoted by

$$I \in \mathbb{R}^{H_I\times W_I\times C}. \tag{2}$$

The goal of a target-image-aware video editing model is to produce an edited video [6]

$$\tilde{V} = \mathcal{F}_\theta(V, I), \tag{3}$$

where $\mathcal{F}_\theta$ is a parameterized mapping that transforms the source video in a way that reflects the appearance or style defined by the target image, while maintaining the motion and scene layout of the original video.

To make the problem more precise, it is useful to distinguish between motion, structure, and appearance components. Let each frame V_t be decomposed conceptually into a structural representation capturing geometry and pose, and an appearance representation capturing texture and identity. While an exact decomposition is not observable, generative models often implicitly separate motion latents from content latents. Assume

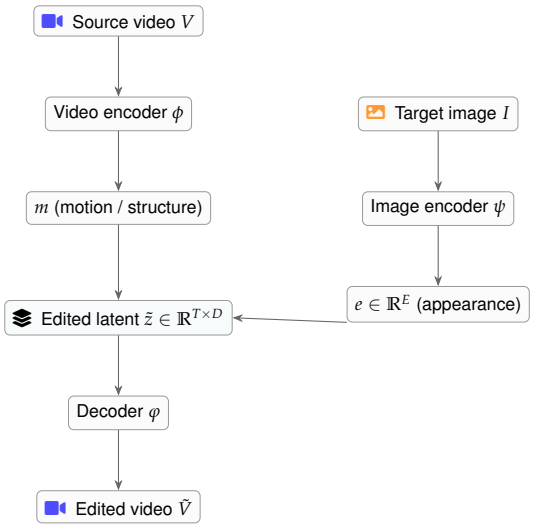

Figure 2 Latent-space view of the conditional transformation. A video encoder ϕ maps the source video V into a motion- and structure-centric latent m, while an image encoder ψ maps the target image I into an appearance embedding e. Both are fused into an edited latent trajectory $\tilde{z}$, which the decoder φ converts into the final edited video $\tilde{V}$, enabling motion preservation and appearance transfer within a shared latent space.

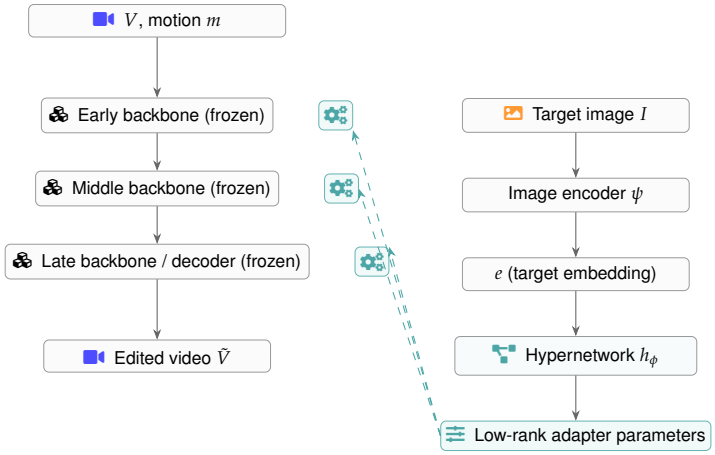

Figure 3 Backbone and adapter architecture for target-image-aware video editing. A pretrained video generator backbone G_{θ_0} processes motion-conditioned inputs, while a target-image embedding e is obtained by encoding I and fed into a hypernetwork h_ϕ that predicts low-rank adapter parameters. These adapters are injected into selected backbone blocks, enabling appearance control via small, data-efficient parameter perturbations without modifying most of the pretrained weights.

there exists a representation

$$z = \phi(V), \tag{4}$$

where z lies in a latent video space

$$z \in \mathbb{R}^{T \times D}, \tag{5}$$

with D denoting the latent dimensionality per frame. Similarly, let the target image be encoded as

$$e = \psi(I), \tag{6}$$

with

$$e \in \mathbb{R}^E. \tag{7}$$

The edited video is then obtained by decoding a modified latent representation

$$\tilde{z} = g_\theta(z, e) \tag{8}$$

through a decoder

$$\tilde{V} = \varphi(\tilde{z}), \tag{9}$$

where ϕ and φ are typically derived from a pretrained video generative model, and g_θ encodes the effect of the target-image-aware edit.

Training data are assumed to consist of tuples of the form [7]

$$(V, I, V^\star), \tag{10}$$

where $V^\star$ is an ideal edited version of V that agrees with the target image I in terms of appearance or style. In practice, perfect target videos may not be available, and supervision may be weak or partially synthetic. Nevertheless, for analysis it is useful to assume that training samples are drawn from a joint distribution

$$(V, I, V^\star) \sim \mathcal{D}, \tag{11}$$

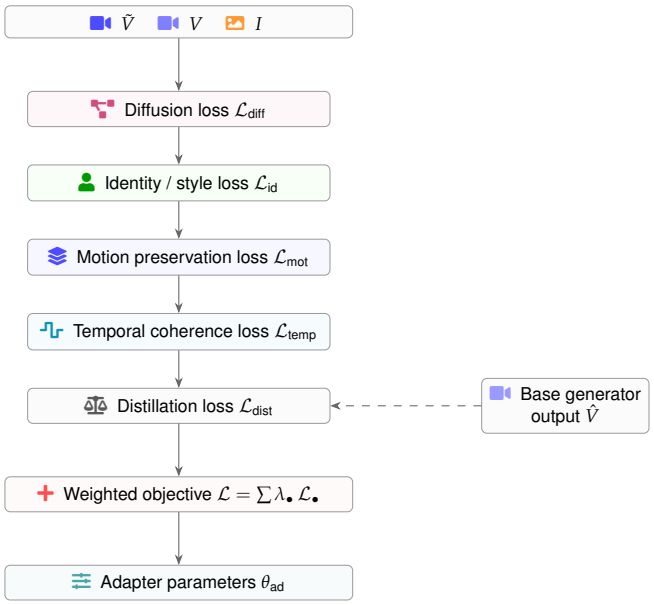

Figure 4 Training objective for data-efficient target-image-aware video editing. Edited videos $\tilde{V}$, source videos V, target images I, and base generator outputs $\hat{V}$ feed into diffusion, identity, motion, temporal, and distillation losses. A weighted combination of these terms defines the overall objective $\mathcal{L}$ that updates only the adapter parameters θ_{ad}, balancing identity transfer, motion preservation, temporal smoothness, and fidelity to the pretrained generator.

and that the objective is to learn parameters θ minimizing the expected editing risk

$$\mathcal{R}(\theta) = \mathbb{E}_{\mathcal{D}}\big[\ell(\mathcal{F}_\theta(V, I), V^\star)\big], \tag{12}$$

for some loss function ℓ that measures discrepancy between the edited and ideal videos.

The loss ℓ should capture multiple desiderata. A basic requirement is framewise fidelity, such as reconstruction error in pixel or latent space. However, this is insufficient, because it does not guarantee identity or style compatibility between $\tilde{V}$ and I, nor does it capture temporal coherence. To model identity or style, one can introduce an embedding function

$$\eta(\cdot) \tag{13}$$

that maps images or video frames into a semantic feature space. An identity or style consistency loss can then be defined as a distance in this feature space between the target image embedding $\eta(I)$ and embeddings of selected frames of $\tilde{V}$. Temporal coherence can be approximated by regularizing differences between consecutive frames or by constraining latent trajectories across time [8].

Data efficiency arises from the fact that the number of available pairs $(V, I, V^\star)$ is limited. In addition, the distribution $\mathcal{D}$ may be skewed: many different source videos may share a small number of target images, or the available pairs may cover only a restricted range of motions and viewpoints. It is therefore advantageous to decompose the model into a pretrained backbone with parameters θ_0 and a smaller set of adaptation parameters $\Delta\theta$, so that

$$\theta = \theta_0 + \Delta\theta, \tag{14}$$

and only $\Delta\theta$ needs to be learned from the limited task-specific data. The backbone captures generic video generation and editing capabilities, while the adaptation parameters encode the specific mechanisms required for target-image-aware behavior.

In many scenarios, there is a second layer of data scarcity at the per-target level. Once a generic model has been trained, it may be adapted to a new target image using only a few examples, such as one or several portrait photographs. Let $\mathcal{T}$ index targets, and let each target τ be associated with a small support set of images

$$\mathcal{S}_\tau = \{I_\tau^{(n)}\}_{n=1}^{N_\tau}. \tag{15}$$

The task is to instantiate a target-specific editor $\mathcal{F}_{\theta,\tau}$ that conditions on $\mathcal{S}_\tau$ and generalizes to arbitrary source videos. In this case, the adaptation parameters become target-dependent, for example

$$\Delta\theta_\tau = h_\phi(\mathcal{S}_\tau), \tag{16}$$

where h_ϕ is a hypernetwork or meta-learner, trained over many targets. The resulting model is evaluated on a distribution over targets and source videos, and data efficiency becomes dependent on both the number of targets seen during training and the number of examples per target.

This formulation naturally connects to a probabilistic view. Assume that videos are generated from a latent process with motion latents m and appearance latents a. Let

$$p(m, a, V) = p(m)\, p(a)\, p(V \mid m, a) \tag{17}$$

denote a generative model, with corresponding approximate posterior encoders for (m, a) given V [9]. Target-image-aware editing can be viewed as inferring the motion latents m from the source video, inferring appearance latents a_I from the target image, and then generating $\tilde{V}$ from $p(V \mid m, a_I)$. In this view, data-efficient training must exploit the prior structure in $p(m)$ and $p(a)$ and the shared decoder $p(V \mid m, a)$ across

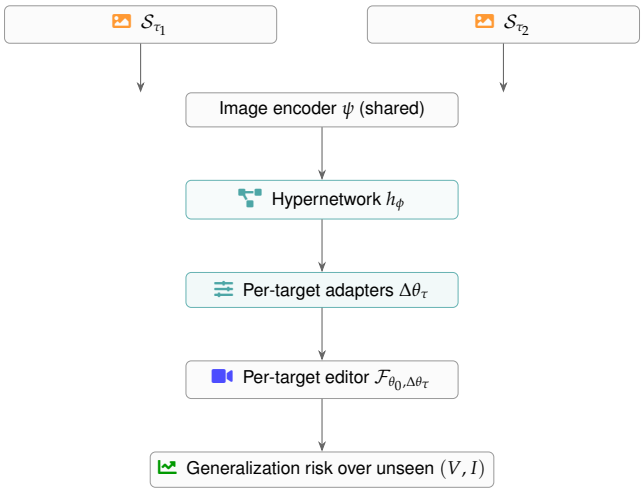

Figure 5 Per-target adaptation and generalization via hypernetworks. Small support sets $\mathcal{S}_\tau$ of target images are encoded by a shared image encoder ψ, and a hypernetwork h_ϕ maps the resulting embeddings to low-dimensional adapter parameters $\Delta\theta_\tau$. These adapters specialize a frozen backbone into per-target editors $\mathcal{F}_{\theta_0,\Delta\theta_\tau}$, whose generalization is evaluated on unseen source videos and target images under strict per-target data budgets.

many editing tasks, while adapting inference and conditioning mechanisms with limited data.

Within this probabilistic framework, the risk $\mathcal{R}(\theta)$ can be decomposed into contributions from errors in motion preservation, identity transfer, and temporal coherence. Let a metric for motion discrepancy be

$$d_{\text{mot}}(V, \tilde{V}), \tag{18}$$

and let a metric for appearance discrepancy relative to I be

$$d_{\text{app}}(I, \tilde{V}). \tag{19}$$

The expected risk can then be expressed as

$$\mathcal{R}(\theta) = \mathbb{E}_{\mathcal{D}}\left[\lambda_{\text{mot}}\, d_{\text{mot}}(V, \tilde{V}) + \lambda_{\text{app}}\, d_{\text{app}}(I, \tilde{V}) + \lambda_{\text{temp}}\, d_{\text{temp}}(\tilde{V})\right], \tag{20}$$

where d_{temp} penalizes temporal inconsistency and the λ coefficients balance the terms. Data-efficient estimation of θ or its adapted variants requires controlling model capacity, regularizing appropriately, and exploiting the structure of these metrics.

In summary, the problem involves learning a conditional transformation $\mathcal{F}_\theta(V, I)$ that is compatible with both target-image features and source video dynamics, subject to limited editing supervision and constrained per-target examples. The remainder of the paper discusses model architectures, objectives, and theoretical properties that address this challenge.

3. Target-Image-Aware Video Editing Model

This section describes a generic architecture for target-image-aware video editing that builds upon a pretrained video generator. The backbone is assumed to be a latent diffusion or autoregressive model that maps noise or latent trajectories to videos, conditioned on text or visual prompts [10]. Target-image awareness is introduced by adding conditioning pathways and parameter-efficient adapters that incorporate target-image embeddings into intermediate representations.

Let the backbone video generator be represented as a mapping

$$G_{\theta_0}(z, c) \tag{21}$$

from a latent trajectory z and conditioning vector c to an output video. In a diffusion setting, z represents a noisy latent at a given time step, and G_{θ_0} predicts noise or denoised latents. In an autoregressive setting, z encodes previous frames or tokens. The conditioning c may include text embeddings and features of the source video. The edited video is obtained by running the generative process with modified conditioning informed by the target image.

The target image I is encoded using a vision encoder yielding a feature vector

$$e = \psi(I) \in \mathbb{R}^E. \tag{22}$$

To influence the generator, e is injected at multiple layers through cross-attention and feature-wise modulation. Consider an intermediate activation tensor [11]

$$X \in \mathbb{R}^{T \times N \times D}, \tag{23}$$

where N is the number of spatial tokens or positions and D is the channel dimensionality. Cross-attention with keys and values derived from e can be applied as

$$Y = X + \text{Attn}\left(XW^Q, eW^K, eW^V\right), \tag{24}$$

where W^Q, W^K, and W^V are learned projection matrices. To be data-efficient, these projections can be decomposed into a frozen component and a low-rank adaptation, for example

$$W^Q = W_0^Q + A^Q B^Q, \tag{25}$$

with A^Q and B^Q of low rank. Analogous decompositions are applied to W^K and W^V. Only the low-rank components are updated during editing-specific training, which limits the number of free parameters and helps avoid overfitting.

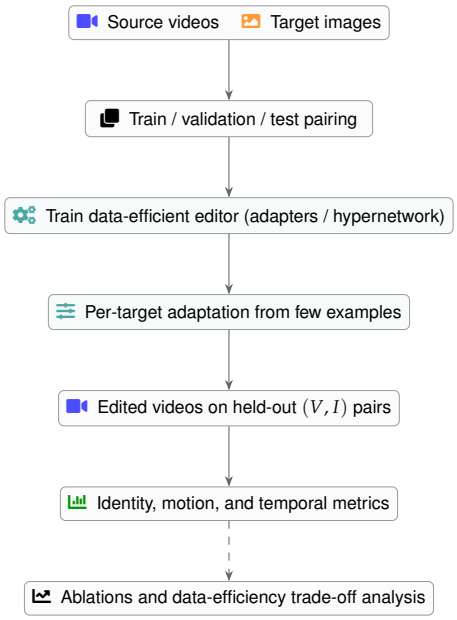

Figure 6 Experimental protocol for assessing data-efficient target-image-aware video editing. Source videos and target images are split into training, validation, and test sets. A backbone with parameter-efficient adapters or hypernetwork conditioning is trained and then adapted to individual targets, after which edited videos are generated for held-out pairs. Identity, motion, and temporal metrics, together with ablation studies, quantify trade-offs between adaptation speed, parameter count, and editing quality under varying data budgets.

In addition to cross-attention, feature-wise modulation conditioned on e can be used. Let a layer normalization output be denoted by

$$\hat{X} = \mathrm{LN}(X). \tag{26}$$

A modulation function

$$(\gamma, \beta) = f_{\theta_{\mathrm{mod}}}(e) \tag{27}$$

produces scale and shift vectors, and the modulated activation is

$$\tilde{X} = \gamma \odot \hat{X} + \beta, \tag{28}$$

where $\odot$ denotes element-wise multiplication [12]. The parameters θ_{mod} are again implemented through small matrices or multi-layer perceptrons with low-rank weight updates. Temporal structure is preserved by sharing modulation parameters across time steps or by adding a simple dependence on frame indices.

To ensure that motion is primarily determined by the source video, a motion encoder processes the source sequence into a latent representation

$$m = \phi_{\mathrm{mot}}(V) \in \mathbb{R}^{T \times D_m}. \tag{29}$$

This representation enters the generator as part of the conditioning c, either by initializing the latent trajectory z, by injecting tokens into attention layers, or by modulating early layers of the backbone. The key idea is that motion latents are derived from the source video and remain fixed when changing the target image, while the appearance signal comes from e.

The edited video is produced by running the generative process with the combination of motion and appearance signals. In a diffusion model, one uses a forward process that maps an initial latent z_0 to noisy latents z_t, and a reverse process parameterized by a denoising network

$$\epsilon_\theta(z_t, c_t), \tag{30}$$

where c_t includes motion and target-image features. The edited latent trajectory is given by

$$z_{t-1} = \alpha_t z_t + \sigma_t \epsilon_\theta(z_t, c_t), \tag{31}$$

with schedule coefficients α_t and σ_t. Only the adapter portions of θ are modified to implement target-image awareness, and the majority of θ_0 is kept frozen.

Per-target adaptation can be performed either by optimizing the adapter parameters directly or by predicting them from the target image via a hypernetwork. In the direct optimization setting, a set of adapter parameters $\Delta\theta$ is initialized to zero and updated using a small number of gradient steps on losses derived from the support set $\mathcal{S}_\tau$. This corresponds to solving an optimization problem

$$\Delta\theta_\tau = \arg\min_{\Delta\theta} \frac{1}{N_\tau} \sum_{n=1}^{N_\tau} L\big(\mathcal{F}_{\theta_0+\Delta\theta}(V^{(n)}, I_\tau^{(n)}), T^{(n)}\big) + \Omega(\Delta\theta), \tag{32}$$

where $T^{(n)}$ denotes target outputs (possibly synthetic) and Ω is a regularizer. The storage cost for per-target adapters can be kept small by restricting $\Delta\theta$ to low-rank or block-diagonal structures [13].

In the hypernetwork setting, a shared network

$$h_\phi : \mathbb{R}^E \to \mathbb{R}^P \tag{33}$$

maps the target embedding e to a vector of adapter parameters of length P. This vector is reshaped into low-rank matrices or

Table 3 Risk decomposition and discrepancy terms for motion, appearance, and time.

Component	Symbol	Description	Examples
Editing risk	$\mathcal{R}(\theta)$	Expected loss over triplets $(V, I, V^\star)$.	Empirical risk over training distribution $\mathcal{D}$.
Motion discrepancy	$d_{\text{mot}}(V, \tilde{V})$	Deviation between source and edited motion.	Flow endpoint error, pose distance, latent motion features.
Appearance discrepancy	$d_{\text{app}}(I, \tilde{V})$	Mismatch between target and edited appearance.	Embedding distance using η_{img}.
Temporal inconsistency	$d_{\text{temp}}(\tilde{V})$	Penalty for frame-to-frame instability.	Difference of increments $\tilde{V}_t - \tilde{V}_{t-1}$.
Balancing weights	$\lambda_{\text{mot}}, \lambda_{\text{app}}, \lambda_{\text{temp}}$	Trade-off between motion, appearance, and smoothness.	Tuned per dataset or application.

small convolution kernels inserted into the generator. The edited video is then obtained as

$$\tilde{V} = \mathcal{F}_{\theta_0, h_\phi}(V, I), \tag{34}$$

where θ_0 is fixed and the conditioning pathway is determined by $h_\phi(e)$. Training h_ϕ across many targets encourages the model to learn a smooth parameter manifold where small changes in e correspond to coherent variations in appearance.

From a tensor calculus perspective, the generator implements a composition of multilinear maps and nonlinearities. Let the sequence of layer transformations be denoted by

$$X^{(l+1)} = \sigma^{(l)}\big(\mathcal{T}^{(l)}(X^{(l)}; \theta^{(l)}, e, m)\big), \tag{35}$$

where $\mathcal{T}^{(l)}$ is a multilinear operator acting on the tensor $X^{(l)}$ and incorporating conditioning via cross-attention and modulation. The target-image-aware adaptation modifies $\mathcal{T}^{(l)}$ through small perturbations in a subspace parameterized by adapters. If the backbone satisfies approximate linearity in a neighborhood of typical activations, then the effect of the adapters can be approximated as a linear functional of e, which connects to the hypernetwork formulation.

This modeling framework separates appearance control from motion preservation, constrains the number of trainable parameters through low-rank and hypernetwork structures, and leverages a pretrained generator to reduce the data needed for effective editing. The next section describes training objectives designed to enforce data-efficient learning, identity transfer, and temporal consistency [14].

4. Data-Efficient Training Objectives

The training objectives for target-image-aware video editing must simultaneously encourage adherence to the target image appearance, preservation of source video motion and structure, and temporal smoothness, while limiting overfitting when the amount of supervision is small. This section describes a set of loss components and regularization strategies that aim to achieve these goals.

A central component in diffusion-based editors is a denoising objective defined in latent space. Let z_0 be the latent encoding of either the ideal edited video $V^\star$ or a proxy constructed via pseudo-labeling. The diffusion forward process adds noise according to

$$z_t = \alpha_t z_0 + \sigma_t \epsilon, \tag{36}$$

where ϵ is sampled from a standard Gaussian. The denoising network ϵ_θ is trained to predict ϵ given z_t, motion-conditioning features, and the target-image embedding. The corresponding loss is

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{z_0, \epsilon, t}\big[\|\epsilon - \epsilon_\theta(z_t, c_t)\|_2^2\big], \tag{37}$$

where c_t aggregates motion and appearance conditioning. When the base generator is already trained, this objective is restricted to adapter parameters and can be further regularized by constraining the deviation of ϵ_θ from the base network.

To enforce fidelity to the target image in terms of identity or style, a perceptual loss in an embedding space is used. Let η_{img} denote a feature extractor applied to images or frames. For a selected subset of frames $\tilde{V}_t$ from the edited video, one defines

$$\mathcal{L}_{\text{id}} = \mathbb{E}\big[\|\eta_{\text{img}}(I) - \eta_{\text{img}}(\tilde{V}_t)\|_2^2\big], \tag{38}$$

where the expectation is over frame indices and data samples. This loss contributes to aligning the appearance of the edited frames with that of the target image, with robustness to pose and lighting variations encoded by the feature extractor [15].

Motion preservation and structural consistency can be addressed by penalizing deviations between the source and edited videos in a motion-sensitive feature space. Let a motion encoder η_{mot} map a video to a sequence of motion features. The motion loss is then

$$\mathcal{L}_{\text{mot}} = \mathbb{E}\big[\|\eta_{\text{mot}}(V) - \eta_{\text{mot}}(\tilde{V})\|_2^2\big], \tag{39}$$

which encourages the edited sequence to follow the same motion trajectory as the source. This loss can be computed on optical

Table 4 Main architectural components of the target-image-aware video editor.

Module	Input	Output	Function
Backbone generator	z, c	Video $G_{\theta_0}(z, c)$	Pretrained video diffusion or autoregressive model.
Cross-attention block	X, e	$Y = X + \mathrm{Attn}(XW^Q, eW^K, eW^V)$	Injects target features into intermediate activations.
Modulation adapter	$\hat{X}, e$	$\tilde{X} = \gamma(e) \odot \hat{X} + \beta(e)$	Feature-wise scale-and-shift conditioned on target image.
Motion conditioning path	V	$m = \phi_{\mathrm{mot}}(V)$	Encodes source motion for conditioning across frames.
Hypernetwork	e	$\Delta\theta_\tau = h_\phi(e)$	Predicts adapter parameters from a target embedding.

flow, skeletal keypoints, or latent features obtained from the pretrained generator, as long as it captures temporal evolution of the scene.

Temporal smoothness can be promoted by penalties on consecutive frame differences in pixel or latent space. For example, a simple temporal regularization term is

$$\mathcal{L}_{\mathrm{temp}} = \mathbb{E}\left[\sum_{t=2}^{T} \| \tilde{V}_t - \tilde{V}_{t-1} - (V_t - V_{t-1}) \|_2^2\right], \tag{40}$$

which constrains temporal increments of the edited video to be similar to those of the source video. This encourages the editing transformation to vary slowly in time and to inherit the temporal structure of the original sequence.

To control overfitting in low-data regimes, it is useful to regularize the edited outputs towards those of the base generator when the edit should be mild [16]. Let $\hat{V}$ denote the output of the base model when conditioned only on the source video and possibly a neutral prompt. A distillation loss can be defined as

$$\mathcal{L}_{\mathrm{dist}} = \mathbb{E}\big[d_{\mathrm{feat}}(\Phi(\tilde{V}), \Phi(\hat{V}))\big], \tag{41}$$

where Φ is a feature extractor and d_{feat} is a distance in feature space. This term encourages the adapters to preserve the general distributional characteristics of the base model, deviating from it only where necessary to satisfy appearance constraints imposed by the target image.

The overall training objective combines these components:

$$\mathcal{L} = \lambda_{\mathrm{diff}}\mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{id}}\mathcal{L}_{\mathrm{id}} + \lambda_{\mathrm{mot}}\mathcal{L}_{\mathrm{mot}} + \lambda_{\mathrm{temp}}\mathcal{L}_{\mathrm{temp}} + \lambda_{\mathrm{dist}}\mathcal{L}_{\mathrm{dist}}, \tag{42}$$

with nonnegative coefficients that balance the influence of each term. Hyperparameters can be tuned to trade off identity fidelity against motion preservation and smoothness.

From a numerical optimization standpoint, training with limited data requires careful conditioning of the gradients. The low-rank adapter structure plays an important role: if the adapter parameters are collected in a vector θ_{ad} and the loss gradient is denoted by $\nabla_{\theta_{\mathrm{ad}}}\mathcal{L}$, then the update step can be written as

$$\theta_{\mathrm{ad}}^{(k+1)} = \theta_{\mathrm{ad}}^{(k)} - \eta\, H^{-1}\nabla_{\theta_{\mathrm{ad}}}\mathcal{L}, \tag{43}$$

where η is a learning rate and H is a preconditioning matrix or an approximation of the Hessian. Low-rank structures imply that θ_{ad} lives in a subspace of the full parameter space, which can reduce the effective condition number of the optimization problem and make the training more stable under noise.

At the per-target adaptation level, where only a few examples are available, the objective is typically restricted to identity and regularization losses, and synthetic reconstruction targets may be used [17]. For a new target τ with support set $\mathcal{S}_\tau$, a meta-learned initialization of adapter parameters is refined by minimizing

$$\mathcal{L}_\tau = \mathbb{E}_{I\in\mathcal{S}_\tau}\big[\mathcal{L}_{\mathrm{id}}(\mathcal{F}_{\theta_{\mathrm{meta}}}(V_{\mathrm{ref}}, I), I)\big] + \Omega(\theta_{\mathrm{ad}}), \tag{44}$$

where V_{ref} is a generic reference video or a short sequence associated with the target. The regularizer Ω can be chosen as an ℓ_2 norm, a nuclear norm of low-rank matrices, or a spectral norm penalty that limits the Lipschitz constant of the adapters. These choices impact the generalization properties, as analyzed in the next section.

From the perspective of probability and statistics, the training objective can be interpreted as minimizing an empirical risk plus a regularization term that encodes prior beliefs about adapter parameters. If $\pi(\theta_{\mathrm{ad}})$ is a prior density and $\mathcal{L}$ is proportional to the negative log-likelihood, the posterior is

$$p(\theta_{\mathrm{ad}} \mid \mathrm{data}) \propto \exp\big(-\mathcal{L}(\theta_{\mathrm{ad}})\big)\pi(\theta_{\mathrm{ad}}). \tag{45}$$

Table 5 Adapter and parameterization strategies for data-efficient conditioning.

Strategy	Parameterization	Advantages	Limitations
Full fine-tuning	Update all θ	Maximum capacity and flexibility.	High sample complexity and storage cost.
Low-rank adapters	$\Delta W = AB$ with small rank	Strong parameter efficiency; regularizes updates.	Rank choice controls expressiveness; too low can underfit.
Shared adapters	Layer-shared $\Delta\theta$	Compact per-target representation.	May limit layer-specific specialization.
Per-target optimization	Optimize $\Delta\theta_\tau$ on $\mathcal{S}_\tau$	Strong personalization from few examples.	Requires optimization per target.
Hypernetwork-generated	$\Delta\theta_\tau = h_\phi(e)$	Instant adaptation; no per-target gradients.	Needs many targets to train h_ϕ robustly.

Data-efficient training benefits from informative priors that favor small adaptations and smooth functional dependence on the target embedding. Bayesian interpretations provide insight into how the number of samples influences posterior concentration and how regularization affects uncertainty in the edited outputs.

Overall, the combination of diffusion-based reconstruction losses, feature-level identity and motion terms, temporal smoothness, and distillation-based regularization provides a flexible framework for data-efficient training and adaptation of target-image-aware video editors. The next section examines theoretical aspects of generalization and sample complexity under this framework.

5. Adaptation and Generalization Analysis

To better understand data efficiency in target-image-aware video editing, it is useful to analyze the relationship between model capacity, regularization, and the number of training samples required for good generalization. This section provides a qualitative and quantitative discussion using simplified models based on linearization and standard tools from statistical learning theory.

Consider a dataset of editing triplets $(V_i, I_i, V_i^\star)$ for $i = 1, \ldots, n$, and let the editing model be parameterized as $\theta = \theta_0 + \theta_{\text{ad}}$, where θ_0 is fixed and θ_{ad} lies in a subspace corresponding to low-rank adapters. For small perturbations relative to θ_0, one can approximate the mapping from parameters to outputs via a first-order Taylor expansion. Let $f_\theta(x)$ denote the edited video for an input pair $x = (V, I)$. Around θ_0,

$$f_\theta(x) \approx f_{\theta_0}(x) + J_{\theta_0}(x)\theta_{\text{ad}}, \tag{46}$$

where $J_{\theta_0}(x)$ is the Jacobian of $f_\theta(x)$ with respect to θ, evaluated at θ_0. Under this linearization, learning θ_{ad} reduces to linear regression in a high-dimensional feature space defined by $J_{\theta_0}(x)$.

Assume a quadratic loss [18]

$$\ell(f_\theta(x), y) = \|f_\theta(x) - y\|_2^2, \tag{47}$$

and define the empirical risk

$$\hat{\mathcal{R}}(\theta_{\text{ad}}) = \frac{1}{n}\sum_{i=1}^{n} \ell(f_\theta(x_i), y_i). \tag{48}$$

In the linearized regime, one can express

$$\hat{\mathcal{R}}(\theta_{\text{ad}}) \approx \frac{1}{n}\sum_{i=1}^{n} \|r_i - J_i\theta_{\text{ad}}\|_2^2, \tag{49}$$

where

$$r_i = y_i - f_{\theta_0}(x_i) \tag{50}$$

and

$$J_i = J_{\theta_0}(x_i). \tag{51}$$

This is a standard linear least squares problem with design matrix formed by stacking the Jacobians J_i. Regularization corresponds to adding a penalty such as

$$\lambda\|\theta_{\text{ad}}\|_2^2, \tag{52}$$

which yields ridge regression.

Classical results for ridge regression imply that the expected generalization error is controlled by the effective dimension of the parameter space, often related to the trace of the kernel matrix or the rank of the design matrix. If the adapter parameters have dimension p but the data live in a subspace where the Jacobian has rank $r \leq p$, then the effective number of degrees of freedom is at most r. Low-rank adapter structures explicitly constrain p by factorizing weight updates into matrices of rank k, such as

$$\Delta W = UV^\top, \tag{53}$$

Table 6 Training objectives combining diffusion, identity, motion, temporal, and distillation terms.

Loss	Definition (sketch)	Intuition
Diffusion denoising	$\mathcal{L}_{\text{diff}} = \mathbb{E}_{z_0,\epsilon,t}\|\epsilon - \epsilon_\theta(z_t, c_t)\|_2^2$	Aligns edited denoising predictions with noisy latent supervision.
Identity/style	$\mathcal{L}_{\text{id}} = \mathbb{E}\|\eta_{\text{img}}(I) - \eta_{\text{img}}(\tilde{V}_t)\|_2^2$	Enforces agreement between target image and edited frames.
Motion preservation	$\mathcal{L}_{\text{mot}} = \mathbb{E}\|\eta_{\text{mot}}(V) - \eta_{\text{mot}}(\tilde{V})\|_2^2$	Encourages the edited video to follow original motion.
Temporal smoothness	$\mathcal{L}_{\text{temp}} = \mathbb{E}\sum_t \|\tilde{V}_t - \tilde{V}_{t-1} - (V_t - V_{t-1})\|_2^2$	Matches temporal increments of source and edited sequences.
Distillation	$\mathcal{L}_{\text{dist}} = \mathbb{E}\, d_{\text{feat}}(\Phi(\tilde{V}), \Phi(\hat{V}))$	Regularizes outputs towards the base generator when edits are mild.
Adapter regularizer	$\Omega(\theta_{\text{ad}})$	Controls adapter norm, rank, or spectrum to reduce overfitting.

with $U \in \mathbb{R}^{d_1 \times k}$ and $V \in \mathbb{R}^{d_2 \times k}$. The number of free parameters in ΔW is then

$$k(d_1 + d_2), \tag{54}$$

instead of [19]

$$d_1 d_2. \tag{55}$$

By choosing k moderately small relative to the layer sizes, one can significantly reduce the capacity of the adapters and thus the sample complexity.

A more refined analysis uses the notion of Rademacher complexity. Let $\mathcal{F}$ denote the class of functions realized by the adapter family, and let $\mathcal{R}_n(\mathcal{F})$ be its empirical Rademacher complexity on a given sample. Generalization bounds typically take the form

$$\mathcal{R}(\theta_{\text{ad}}) - \hat{\mathcal{R}}(\theta_{\text{ad}}) \le \mathcal{O}\Big(\frac{\mathcal{R}_n(\mathcal{F})}{\sqrt{n}}\Big), \tag{56}$$

with high probability. For linear models with bounded parameter norm and bounded input features, the Rademacher complexity scales like

$$\mathcal{R}_n(\mathcal{F}) \le \frac{BR}{\sqrt{n}}, \tag{57}$$

where B bounds the parameter norm and R bounds the feature norms. In the present context, the feature vectors are the Jacobian rows and can be influenced by the choice of normalization and activation functions in the generator. Regularization terms, including spectral norm penalties on adapter matrices, contribute to reducing B and thus the generalization gap.

From a multivariate calculus viewpoint, the impact of adapters on outputs can be quantified by gradients with respect to target-image embeddings [20]. Let the edited output be $\tilde{V} = f_{\theta_{\text{ad}}}(V, I)$ and consider the gradient

$$\frac{\partial \tilde{V}}{\partial e}, \tag{58}$$

where $e = \psi(I)$. This gradient measures sensitivity of the edited video to perturbations of the target embedding. If ψ is fixed, the main contributor to this sensitivity is the adapter mapping from e to layer modulations or weight updates. Bounding the operator norm of this mapping limits the Lipschitz constant of the overall editor with respect to e, which in turn constrains how rapidly edits can vary with changes in the target image. Such bounds are useful when analyzing stability under noise or slight variations in target images.

One can also adopt a probabilistic view based on Bayesian linear regression in the adapter space. Assume the linearized model

$$r = J\theta_{\text{ad}} + \varepsilon, \tag{59}$$

where r stacks the residuals r_i, J stacks the Jacobians, and ε is Gaussian noise with covariance

$$\sigma^2 I. \tag{60}$$

Let the prior on θ_{ad} be Gaussian with covariance

$$\Sigma_0. \tag{61}$$

The posterior covariance then satisfies

$$\Sigma^{-1} = \Sigma_0^{-1} + \frac{1}{\sigma^2} J^\top J. \tag{62}$$

The trace of Σ encodes the remaining uncertainty about adapter parameters after observing data [21]. As the number of samples increases, the term

$$J^\top J \tag{63}$$

dominates and the posterior contracts. Low-rank constraints and strong priors make Σ_0 small in directions orthogonal to the adapter subspace, focusing learning on a limited number of directions where data provide substantial information.

Table 7 Representative data regimes and adaptation setups for target-image-aware editing.

Regime	Supervision	Adaptation mechanism	Typical use case
Global training	Many targets and paired $(V, I, V^\star)$	Learn shared adapters or hypernetwork h_ϕ.	Building a general-purpose editor.
Few-shot per target	Few pairs per τ	Optimize $\Delta\theta_\tau$ with strong regularization.	Personalization to a specific identity.
One-shot target image	Single I_τ, limited video	Use hypernetwork from e; minimal fine-tuning.	Quick user-specific editing.
Weak or synthetic labels	Pseudo $V^\star$ or partial cues	Combine reconstruction, identity, and motion losses.	When perfect edited videos are unavailable.
Cross-domain deployment	Training and test domains differ	Rely on distillation and priors from θ_0.	Applying the editor in-the-wild.

Table 8 Key theoretical quantities relating adapter capacity and generalization.

Aspect	Key quantities	Interpretation
Linearization	Jacobian $J_{\theta_0}(x)$	Describes how small adapter changes affect edited outputs.
Low-rank capacity	Rank k, size of U, V in $\Delta W = UV^\top$	Controls effective dimensionality and sample complexity.
Complexity measure	Rademacher complexity $\mathcal{R}_n(\mathcal{F})$	Bounds generalization gap given bounded adapter norms.
Bayesian view	Posterior covariance Σ with prior Σ_0	Quantifies uncertainty in adapter parameters after observing data.
Generalization bound	Dimension p, samples n, norm bound B	Error scales like $\mathcal{O}(LB\sqrt{p/n})$ under standard assumptions.

Per-target adaptation can be viewed through the lens of meta-learning. Suppose there are many targets τ, each with its own adapter parameters $\theta_{\mathrm{ad},\tau}$ and data sampled from a target-specific distribution $\mathcal{D}_\tau$. A meta-learner seeks a shared initialization or hypernetwork parameters that facilitate rapid adaptation across tasks. Under a bilevel formulation, the meta-objective is

$$\min_{\phi} \ \mathbb{E}_\tau \Big[\mathcal{R}_\tau \big(\theta^\star_{\mathrm{ad},\tau}(\phi) \big) \Big], \tag{64}$$

where

$$\theta^\star_{\mathrm{ad},\tau}(\phi) \tag{65}$$

denotes the result of inner-loop adaptation starting from initialization specified by ϕ. Approximate analysis linearizes the adaptation dynamics and treats them as a gradient step in a quadratic loss landscape. The convergence rate and final error depend on the conditioning of the Hessian and the alignment between the meta-initialization and the per-task optima. Strong regularization and low-rank constraint can improve conditioning, allowing effective adaptation from a small number of gradient steps and few samples per target [22].

Finally, it is instructive to consider a simplified generalization bound that relates adapter capacity and data requirements. Suppose the adapter parameter vector has dimension p and is constrained to satisfy

$$\|\theta_{\mathrm{ad}}\|_2 \le B. \tag{66}$$

Assume the loss function is L-Lipschitz in its first argument and bounded by C. Under standard assumptions, one can derive with high probability a bound of the form

$$\mathcal{R}(\theta_{\mathrm{ad}}) \le \hat{\mathcal{R}}(\theta_{\mathrm{ad}}) + \mathcal{O}\Big(\frac{LB\sqrt{p}}{\sqrt{n}} \Big) + \mathcal{O}\Big(\sqrt{\frac{\log(1/\delta)}{n}} \Big), \tag{67}$$

for failure probability δ. This expression suggests that, for fixed n, decreasing p via low-rank adapters or other parameter-efficient mechanisms reduces the generalization gap. In practice,

Table 9 Experimental factors and typical qualitative trends observed in evaluation.

Factor	Variations	Observed trend
Adapter rank	Small vs. large k in low-rank updates	Higher rank improves fidelity but risks overfitting with few pairs.
Data budget	Number of editing pairs per target	Performance degrades gracefully with parameter-efficient adapters.
Loss weights	$\lambda_{\text{id}}, \lambda_{\text{mot}}, \lambda_{\text{temp}}, \lambda_{\text{dist}}$	Shifts trade-off between identity accuracy and motion preservation.
Temporal regularization	With vs. without $\mathcal{L}_{\text{temp}}$	Removing smoothness increases flicker and inter-frame jitter.
Distillation usage	With vs. without $\mathcal{L}_{\text{dist}}$	Distillation improves robustness and keeps outputs close to backbone distribution.

the effective dimension is typically smaller than p, but the bound highlights the importance of constraining model capacity under data scarcity.

While these analyses rely on idealized assumptions, such as linearization and Gaussian noise, they indicate that data-efficient training and adaptation of target-image-aware video editors benefit from low-rank parameterizations, strong regularization, and carefully controlled sensitivity to target embeddings. They also suggest that meta-learning of hypernetworks can distribute the cost of learning across many targets, reducing per-target sample complexity [23].

6. Experimental Evaluation

Empirical evaluation of data-efficient target-image-aware video editing models involves several aspects: the quality and consistency of edits under varying data budgets, the trade-off between identity fidelity and motion preservation, and the impact of different adapter and regularization strategies. This section outlines a typical experimental setup, metrics, and observations that can be made when investigating such models.

A common experimental protocol considers a collection of source videos spanning a variety of scenes, motions, and lighting conditions, together with a set of target identities or styles represented by images. For each target, a small number of reference images is provided, ranging from a single image to a few diverse views. The model is trained or adapted on a subset of source videos paired with target images, and evaluated on held-out videos and targets. Different data regimes are studied by varying the number of editing pairs per target and the number of distinct targets seen during training.

Evaluation requires metrics that capture the multi-faceted nature of the task. Identity or style transfer is assessed by computing distances between embeddings of the target image and frames from the edited video in a feature space trained for recognition or style discrimination [24]. Smaller feature distances indicate better transfer. Motion preservation is evaluated by comparing motion descriptors, such as optical flow or pose trajectories, between source and edited videos. One can compute the average endpoint error or orientation difference of flow vectors, or measure distances between sequences of joint positions in human-centric videos. Temporal consistency can be assessed using frame difference statistics, short-term consistency scores based on warping errors, or coherence metrics that compare features of consecutive frames.

In addition to objective metrics, qualitative inspection of edited videos is important to reveal artifacts such as flickering, identity drift, or background corruption that may not be fully captured by numerical scores. Human preference studies or small-scale user evaluations can provide further insight into perceived quality. However, in data-efficient settings, it is often necessary to rely on automated metrics for large-scale comparisons, since manual inspection of many edited videos would be time-consuming.

Experiments comparing different adapter architectures can highlight the role of parameter efficiency. For example, one can compare full fine-tuning of all generator parameters, low-rank adapters with various ranks, and hypernetwork-generated adapters [25]. Under a fixed training set size, full fine-tuning may achieve high training performance but exhibit overfitting and poor generalization to new videos or targets, manifesting as unstable edits or loss of motion structure. Low-rank adapters with appropriately chosen rank can achieve similar or better generalization with fewer trainable parameters and improved stability, especially when combined with strong regularization. Hypernetwork-based approaches allow sharing of information across targets and can improve performance when the number of distinct targets is large but per-target data is scarce.

A key set of experiments examines the effect of data budget

on performance. By systematically reducing the number of editing pairs per target, one can observe how metrics degrade and how quickly different models lose identity fidelity or temporal consistency. Models that rely heavily on target-specific fine-tuning tend to degrade rapidly as the number of examples decreases, while hypernetwork-based models that have been trained on many targets can maintain reasonable performance even in the one-shot setting. The presence of distillation and temporal regularization terms mitigates overfitting and helps maintain smooth and coherent edits under low data.

Another set of experiments investigates the effect of the weighting coefficients in the loss function. By varying λ_{id}, λ_{mot}, and λ_{temp}, one can move along a continuum between strong identity emphasis and strong motion preservation. High λ_{id} values lead to more faithful reproduction of target appearance but can distort motion or induce artifacts, particularly when the target and source differ substantially. Conversely, large λ_{mot} values may preserve motion but fail to fully transfer the target identity. Selecting a balance that yields visually acceptable results often depends on the application and can be guided by empirical sweeps over the coefficient space [26].

Ablation studies help disentangle the contributions of various components. For instance, removing temporal smoothness regularization while keeping other losses fixed can result in edited videos that accurately match identity on a framewise basis but exhibit inter-frame jitter. Removing distillation tends to increase the diversity of outputs but may also allow deviations from the naturalness of the base generator, sometimes manifesting as unrealistic textures or inconsistent lighting. Eliminating motion-aware losses can cause the editor to alter the scene layout or produce unstable motion patterns, demonstrating the necessity of considering motion explicitly.

In realistic scenarios, the distribution shift between training and deployment data must be considered. Editing models trained on videos with constrained camera motion or limited background complexity may encounter difficulties when applied to unconstrained videos. Experiments that evaluate models on different domains, such as moving from controlled indoor clips to outdoor scenes with complex backgrounds, reveal the robustness of the learned adapters and regularization schemes. Data-efficient approaches that rely on strong priors and distillation often exhibit better robustness, as they retain more of the inductive biases of the pretrained backbone.

Computational efficiency and adaptation time are also important [27]. In per-target adaptation experiments, one can measure the time required to obtain a usable editor for a new target, as a function of the number of gradient steps and the complexity of the adapter. Models with lightweight adapters and hypernetwork conditioning can often adapt faster than those requiring substantial fine-tuning of the backbone. In addition, the memory footprint of storing per-target adapters matters when scaling to many personalized editors. Low-rank or sparse adapter representations reduce storage requirements and make the approach more scalable.

Experiments may also include stress tests that probe model behavior under large pose changes, occlusions, and lighting variations. By selecting source videos with challenging motion patterns or partial occlusions of the edited region, one can assess how well the model can interpolate missing information and maintain identity consistency across frames. Data-efficient models that have learned to rely on stable latent motion representations and robust appearance features tend to perform better in these regimes, whereas models that overfit to specific training configurations may struggle.

In summary, experimental evaluations of data-efficient target-image-aware video editing models focus on how well the models balance identity fidelity, motion preservation, temporal coherence, and robustness under varying data budgets and domain shifts. Comparisons between full fine-tuning, low-rank adapters, and hypernetwork-based adaptation illustrate the advantages of parameter-efficient conditioning and appropriate regularization for achieving stable performance in low-data settings [28].

7. Conclusion

This paper has examined data-efficient training and adaptation strategies for target-image-aware video editing models that operate by transforming source videos according to the appearance or style encoded in a target image. The problem was formulated as learning a conditional mapping from source videos and target images to edited videos, with explicit consideration of motion preservation, identity or style transfer, and temporal coherence. Within this setting, the scarcity of paired editing data and the limited number of examples per target motivate models that exploit pretrained video generators and parameter-efficient conditioning mechanisms.

A model architecture was described in which target-image embeddings are injected into a pretrained video generator via cross-attention and feature-wise modulation, while motion information is extracted from the source video and preserved through separate conditioning pathways. Parameter-efficient adapters, such as low-rank updates and hypernetwork-generated weights, confine learning to a low-dimensional subspace of parameter space, reducing the risk of overfitting and improving data efficiency. The use of adapters makes it possible to balance expressiveness and sample complexity, and to store per-target parameters compactly when personalization is required.

Training objectives for data-efficient video editing were discussed, combining diffusion-based denoising losses, feature-level identity or style consistency terms, motion preservation penalties, temporal smoothness constraints, and distillation from the base generator. These components are designed to encourage edited videos that remain faithful to target appearance while retaining source motion and structural layout, and to do so under limited supervision. Regularization and priors on adapter parameters play an important role, effectively constraining the space of possible edits [29].

A theoretical perspective based on linearization, Rademacher complexity, and Bayesian interpretations was used to analyze generalization properties and sample complexity. Under simplifying assumptions, the analysis indicates that constraining adapter capacity through low-rank structures and norm bounds reduces the number of samples needed to obtain reliable editing performance. Sensitivity to target embeddings and stability of

the editing function are related to operator norms of adapter mappings, suggesting that controlling these norms can improve robustness under variation in target images and noise.

Experimental considerations were outlined, including evaluation metrics for identity fidelity, motion preservation, and temporal coherence, as well as comparisons between different adapter architectures and regularization schemes. The discussion emphasized how data-efficient approaches can maintain reasonable performance under varying data budgets and domain shifts, and highlighted trade-offs between adaptation speed, storage, and editing quality. While concrete numerical results depend on specific implementations and datasets, the outlined framework provides a structured way to investigate and compare data-efficient video editing methods. the study suggests that combining strong pretrained video generators with carefully designed, parameter-efficient conditioning and regularization strategies is a promising route for achieving data-efficient target-image-aware video editing. Future work may explore richer motion representations, improved hypernetwork designs, and integration of additional modalities, such as audio or depth, while maintaining or further improving data efficiency. In addition, more refined theoretical analyses that relax linearization assumptions and better capture the nonlinearity of modern generative models could provide deeper insight into the limits and capabilities of data-efficient adaptation in video editing [30].

Acknowledgments

We would like to thank the reviewers of this research for their helpful comments and suggestions.

References

[1] F. Zhang, J. Zhao, Y. Zhang, and S. Zollmann, "A survey on 360° images and videos in mixed reality: Algorithms and applications," *Journal of Computer Science and Technology*, vol. 38, pp. 473–491, 5 2023.

[2] S. Ma, J. Gao, R. Wang, J. Chang, Q. Mao, Z. Huang, and C. Jia, "Overview of intelligent video coding: from model-based to learning-based approaches," *Visual Intelligence*, vol. 1, 8 2023.

[3] U. B. A, "Blinds personal assistant application for android," *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, pp. 138–144, 1 2019.

[4] M. A. Posicelskaya, T. A. Rudchenko, and A. L. Semenov, "Mathematical elements of elementary education," *Doklady Mathematics*, vol. 107, pp. S10–S41, 8 2023.

[5] S. S. Harsha, A. Revanur, D. Agarwal, and S. Agrawal, "Genvideo: One-shot target-image and shape aware video editing using t2i diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7559–7568, 2024.

[6] E. Salehi, A. Aghagolzadeh, and R. Hosseini, "Stereo-rivo: Stereo-robust indirect visual odometry," *Journal of Intelligent & Robotic Systems*, vol. 110, 7 2024.

[7] Z. Gu, Z. Li, X. Di, and R. Shi, "An lstm-based autonomous driving model using a waymo open dataset," *Applied Sciences*, vol. 10, pp. 2046–, 3 2020.

[8] J. Yu, "Research on key development technologies for spoc platform," *MATEC Web of Conferences*, vol. 232, pp. 01025–, 11 2018.

[9] G. Parasher, M. Wong, and M. Rawat, "Evolving role of artificial intelligence in gastrointestinal endoscopy.," *World journal of gastroenterology*, vol. 26, pp. 7287–7298, 12 2020.

[10] R. Patil, N. V. Goutham, G. R. S. Kumar, and S. Borra, "Alexnet based pirate detection system," *SN Computer Science*, vol. 3, 12 2021.

[11] R. Deepa, T. S. Sharmila, and R. Niruban, "Dynamic graph neural network-based computational paradigm for video summarization," *Multimedia Tools and Applications*, vol. 83, pp. 51227–51250, 11 2023.

[12] G. Sziebig, B. Solvang, and P. Korondi, *Image Processing for Next-Generation Robots*, pp. 429–440. InTech, 11 2008.

[13] N. Shah, N. Bhagat, and M. Shah, "Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention," *Visual computing for industry, biomedicine, and art*, vol. 4, pp. 9–9, 4 2021.

[14] S.-K. Zhang, J.-H. Liu, Y. Li, T. Xiong, K.-X. Ren, H. Fu, and S.-H. Zhang, "Automatic generation of commercial scenes," in *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1137–1147, ACM, 10 2023.

[15] J. Dudley and P. O. Kristensson, "A review of user interface design for interactive machine learning," *ACM Transactions on Interactive Intelligent Systems*, vol. 8, pp. 8–37, 6 2018.

[16] C. Ct, S. Dd, M. Bovyn, S. P. Gross, X. Xie, and Z. S. Siwy, "Deep learning assisted mechanotyping of individual cells through repeated deformations and relaxations in undulating channels," 5 2021.

[17] M. K. Wozniak, R. Stower, P. Jensfelt, and A. Pereira, "What you see is (not) what you get," in *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 243–247, ACM, 3 2023.

[18] B. Buchberger, "Automated programming, symbolic computation, machine learning: my personal view," *Annals of Mathematics and Artificial Intelligence*, vol. 91, pp. 569–589, 10 2023.

[19] S. Agnihotri and A. Agarwal, "Ambient artificial intelligence," *International Journal of Computer Applications*, vol. 170, pp. 1–5, 7 2017.

[20] V. Guan, C. Zhou, H. Wan, R. Zhou, D. Zhang, S. Zhang, W. Yang, B. P. Voutharoja, L. Wang, K. T. Win, and P. Wang, "A novel mobile app for personalized dietary advice leveraging persuasive technology, computer vision, and cloud computing: Development and usability study.," *JMIR formative research*, vol. 7, pp. e46839–e46839, 8 2023.

[21] C. G. Northcutt, S. Zha, S. Lovegrove, and R. Newcombe, "Egocom: A multi-person multi-modal egocentric communications dataset," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, pp. 1–1, 5 2023.

[22] K. Shah, R. Sushra, M. Shah, D. Shah, H. Shah, M. Raval, and M. Prajapati, "Crop protection and disease detection using artificial intelligence and computer vision: a comprehensive review," *Multimedia Tools and Applications*, vol. 84, pp. 3723–3743, 4 2024.

[23] Y. Zhong, L. Sun, C. Ge, and H. Fan, "Hog-esrs face emotion recognition algorithm based on hog feature and esrs method," *Symmetry*, vol. 13, pp. 228–, 1 2021.

[24] I. Wahlang, A. K. Maji, G. Saha, P. Chakrabarti, M. Jasinski, Z. Leonowicz, and E. Jasińska, "Deep learning methods for classification of certain abnormalities in echocardiography," *Electronics*, vol. 10, pp. 495–, 2 2021.

[25] M. D.K., R. R, E. V, and S. A, "An improvised video enhancement using machine learning," *ICTACT Journal on Image and Video Processing*, vol. 13, pp. 2844–2848, 11 2022.

[26] X. Gong and F. Wang, "Classification of tennis video types based on machine learning technology," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–11, 6 2021.

[27] X. Chang, J. Yu, Y. Gao, H. Ding, Y. Liu, and H. Yu, "Classification of sailboat tell tail based on deep learning," *Journal of Ocean University of China*, vol. 23, pp. 710–720, 5 2024.

[28] K. Xia, C. Saidy, M. Kirkpatrick, N. C. Anumbe, A. Sheth, and R. Harik, "Towards semantic integration of machine vision systems to aid manufacturing event understanding.," *Sensors (Basel, Switzerland)*, vol. 21, pp. 4276–, 6 2021.

[29] R. Fan, F.-L. Zhang, M. Zhang, and R. R. Martin, "Robust tracking-by-detection using a selection and completion mechanism," *Computational Visual Media*, vol. 3, pp. 285–294, 5 2017.

[30] G. S. Letterie, "Three ways of knowing: the integration of clinical expertise, evidence-based medicine, and artificial intelligence in assisted reproductive technologies," *Journal of assisted reproduction and genetics*, vol. 38, pp. 1617–1625, 4 2021.