Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

LTX-Insight: Unified Video Restoration and Semantic Editing via Task-Aware Adaptation and Temporal Consistency

Fan Tang

tangfan2021@yeah.net

AI Laboratory, Chongqing Valiant Cat Technology https://orcid.org/0009-0008-7945-8693

Siyuan Li

AI Laboratory, Chongqing Valiant Cat Technology https://orcid.org/0000-0002-2354-4233

Research Article

Keywords: video diffusion, multi-task learning, temporal consistency, video restoration, semantic editing

Posted Date: May 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9775063/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare potential competing interests as follows: The authors declare potential competing interests as follows: Competing Interests Both authors are affiliated with Chongqing Valiant Cat Technology Co., Ltd., which organized this research. The authors declare that they have no additional competing financial or non-financial interests.

LTX-Insight: Unified Video Restoration and Semantic Editing via Task-Aware Adaptation and Temporal Consistency

Fan Tang Siyuan Li
AI Laboratory, Chongqing Valiant Cat Technology Co., Ltd.
tangfan2021@yeah.net, qq497291093@gmail.com
* Correspondence: tangfan2021@yeah.net

Abstract

Real-world video post-processing often involves multiple high-frequency tasks, including video super-resolution, occlusion removal and repair, and instruction-driven semantic editing. However, most existing methods are designed for a single scenario, or rely on frame-by-frame processing and multi-model cascades, which easily lead to task interference, cross-frame flickering, and high deployment cost. To address these issues, we propose **LTX-Insight**, a unified framework for video restoration and semantic editing, which jointly models multiple video post-processing tasks on a shared video diffusion backbone. Specifically, **TADA (Task-Aware Decoupled Adapter)** introduces a task-aware branch-separated adaptation mechanism with a weak decoupling constraint to regulate structure restoration and texture-semantic editing at different layers, thereby alleviating semantic interference under shared multi-task parameters. **FGLC (Flow-Guided Latent Consistency)** introduces explicit temporal consistency constraints in latent space by combining optical-flow alignment and cross-frame attention bias, reducing flickering while improving the stability of motion regions and occlusion completion. **PMTR (Progressive Multi-Task Routing)** enhances convergence stability and generalization under joint multi-task training through progressive curriculum learning and collaborative task routing. We conduct systematic experiments and ablation studies on three tasks, namely video super-resolution, occlusion removal, and semantic editing, validating the effectiveness and complementarity of all modules. Experimental results show that our method consistently improves editability, detail preservation, and visual fidelity over representative baselines.

Keywords: video diffusion; multi-task learning; temporal consistency; video restoration; semantic editing

1 Introduction

With the continued growth of short-video production, AIGC content generation, and digital asset re-editing, the goal of video post-processing systems has shifted from traditional "single-task enhancement" to "unified controllable editing." In real-world scenarios, users often expect a single model to simultaneously support super-resolution, occlusion removal, and semantic editing, rather than repeatedly switching among multiple independent models. This trend imposes higher requirements on model design: it must ensure not only per-frame visual quality, but also cross-frame consistency, controllability, and deployment efficiency.

In recent years, diffusion models have achieved remarkable progress in image and video generation, and spatio-temporal Transformer-based methods have demonstrated strong performance in detail representation and semantic alignment [1, 2, 3]. Nevertheless, most existing studies either focus on single-task optimality or adopt an engineering pipeline of "separate task-wise training followed by cascaded deployment," as commonly observed in video restoration and editing systems [4, 5]. When task types change, such schemes easily suffer from high transfer cost, parameter redundancy, and increased maintenance complexity, making them unsuitable for building a unified video editing framework.

From the perspective of optimization objectives, super-resolution, occlusion removal, and semantic editing are inherently heterogeneous. Super-resolution emphasizes high-frequency texture reconstruction and perceptual quality enhancement; occlusion removal focuses on semantic completion of locally occluded regions and restoration of the surrounding background; semantic editing requires controlled modification of target regions while preserving non-target regions as much as possible. If one uses only a unified global injection strategy or a single LoRA increment, the model often

exhibits task interference, such as excessive background modification, structural boundary drift, and semantic contamination outside the edited region.

In addition to task conflict, temporal stability remains a core bottleneck in video modeling. Many methods can achieve strong frame-wise metrics, yet still suffer from flickering, texture jitter, and motion-region misalignment during continuous playback. Existing video editing studies have shown that cross-frame attention reuse or feature propagation can alleviate such inconsistency to some extent [6, 7]. The underlying reason is that frame-wise losses cannot explicitly constrain cross-frame correspondences. This is particularly severe in occlusion removal and local editing scenarios, where reconstructed occluded regions may vary over time, significantly degrading subjective visual quality. Therefore, building a temporal consistency mechanism in latent space is essential for unified video editing systems.

Furthermore, deployment itself introduces practical constraints. Multi-model parallel solutions usually incur higher memory usage, more complex inference scheduling, and greater maintenance cost, and these issues become more pronounced as resolution and video length increase. Compared with "multi-model stitching," industrial scenarios increasingly require a "single-model multi-task collaboration" paradigm, namely, realizing task-specific representation, temporally stable generation, and scalable training under a shared backbone.

Motivated by the above observations, we propose LTX-Insight, a unified video diffusion framework, and provide corresponding solutions to three key problems: improving cross-task controllability through task-aware hierarchical injection, enhancing temporal consistency via flow-guided latent constraints, and improving joint training stability through progressive multi-task routing. The framework is designed to balance quality, stability, and unification without substantially increasing deployment complexity.

Contributions.

- We propose **TADA (Task-Aware Decoupled Adapter)**, which models the distinct requirements of "structure restoration" and "semantic editing" through task-aware gating, branch-separated injection, and a weak orthogonality constraint, thereby alleviating semantic interference under a shared multi-task backbone.
- We propose **FGLC (Flow-Guided Latent Consistency)**, which explicitly constrains cross-frame consistency in latent space by jointly modeling optical-flow alignment and attention bias, reducing flicker and motion-region reconstruction drift.
- We propose **PMTR (Progressive Multi-Task Routing)**, which improves stable convergence and generalization of single-model multi-task training through curriculum-based task scheduling and collaborative routing optimization.

2 Related Work

2.1 Video Diffusion Generation and Editing

The success of diffusion models in image generation has driven their extension to the video domain. Compared with GANs or autoregressive paradigms, diffusion-based methods are generally more stable in high-fidelity detail reconstruction and complex semantic alignment. Representative works such as Make-A-Video [1] and Imagen Video [2] validate the effectiveness of cascaded diffusion for text-to-video generation, while Video LDM [3] further demonstrates the efficiency advantage of latent-space modeling for high-resolution video synthesis.

In video editing tasks, mainstream methods usually combine textual conditions, reference-frame conditions, or structural conditions (such as edges, depth, or pose) to achieve controllable generation. Tune-A-Video [5] and Text2Video-Zero [8] lower the barrier of video editing from the perspectives of few-shot fine-tuning and zero-shot transfer, respectively, while AnimateDiff [9] improves the usability of transferring image models to video models through plug-and-play motion modules. Although these methods perform well under single-task or specific-condition settings, models may still suffer from insufficient control granularity or content drift when switching from "quality enhancement" to "semantic rewriting," indicating that task decoupling within unified frameworks remains insufficiently addressed.

2.2 Parameter-Efficient Fine-Tuning and Adaptation Mechanisms

Parameter-efficient adaptation has become a key technical route for deploying diffusion models in practice. In the direction of control branches and adapters, ControlNet [10] realizes spatial conditional control through an additional controllable branch, while T2I-Adapter [11] and IP-Adapter [12] enhance controllable generation from the perspectives of structural signals and image prompts, respectively. These methods jointly show that when the backbone is frozen or partially frozen,

introducing lightweight trainable modules can significantly reduce tuning and deployment cost while preserving generative capability.

However, most existing parameter-efficient schemes are biased toward single-task or single-control settings. Even in video scenarios, works such as Tune-A-Video [5] that explore few-shot adaptation still commonly adopt a strategy of "unified injection with unified strength," making it difficult to explicitly distinguish the sensitivity of different layers to structural information versus texture information. When multiple tasks share the same backbone, such indiscriminate injection easily causes task coupling and gradient interference. Therefore, how to realize hierarchical task control while maintaining parameter efficiency remains a core challenge for adaptation methods.

2.3 Temporal Consistency Modeling

Video quality depends not only on per-frame sharpness, but also on cross-frame consistency. Existing temporal modeling methods can be broadly divided into three categories: explicit alignment based on optical flow or motion fields, implicit association based on cross-frame attention, and hybrid methods combining both. Representative works such as FateZero [6], TokenFlow [7], Rerender A Video [13], and VidToMe [14] alleviate flickering in video editing from the perspectives of attention fusion, feature propagation, keyframe translation, and cross-frame token merging, respectively.

Within diffusion frameworks, temporal consistency is even more challenging because the denoising process spans multiple timesteps, and errors may accumulate and amplify gradually. Although existing methods can significantly improve consistency, they still face a trade-off between stability and efficiency in long videos, strong-motion videos, or local reconstruction scenarios. This suggests that it is necessary to build temporal constraint mechanisms in latent space that jointly balance interpretability, robustness, and computational efficiency.

2.4 Joint Multi-Task Training and Routing Collaboration

The goal of multi-task learning is to mine commonality across tasks in shared representations while suppressing negative transfer. In video restoration, VRT [4] demonstrates the potential of unified architectures for multiple restoration tasks. In multi-task representation learning, TaskPrompter [15] improves the collaborative modeling of shared and specialized representations through task prompting. Meanwhile, the sparse-expert paradigm based on conditional routing, as verified in ST-MoE [16] and GLaM [17], shows the scalability of a "shared backbone + conditional activation" design.

However, for video editing scenarios, large differences in task objectives, imbalanced sample distributions, and inconsistent convergence rates make "one-shot mixed training" still prone to main-task bias or underfitting of weaker tasks. Although the joint design of curriculum learning and routing mechanisms is promising, systematic solutions are still lacking in unified video restoration and semantic editing, especially under the single-model constraint where quality, temporal consistency, and controllability must all be considered simultaneously.

3 Method

3.1 System Overview and Training Pipeline

Given an input video $X = \{x_i\}_{i=1}^{T}$, a text instruction p, and a task label $r \in \{\text{upscale}, \text{occlusion_rm}, \text{edit}\}$, our goal is to learn a conditional mapping from the input video to the target video $Y^* = \{y_i^*\}_{i=1}^{T}$ within a single video diffusion backbone. The external input format is the same for all three tasks: a source video, a natural-language prompt, and a task label. No explicit occlusion or editing mask is required at inference time. Rather than training separate models for super-resolution, occlusion removal, and semantic editing, LTX-Insight unifies the three tasks as a conditional latent-space video generation problem, and introduces lightweight task-specific modules to mitigate negative transfer in shared multi-task representations.

The overall method consists of three mutually complementary modules. First, TADA (Task-Aware Decoupled Adapter) introduces task-aware gating at different layers of the spatio-temporal Transformer, allowing structure restoration and texture/semantic editing to be handled through branch-separated residual pathways with an explicit weak decoupling regularizer, thereby reducing excessive coupling caused by a single LoRA or a unified adapter under multi-task settings. Second, FGLC (Flow-Guided Latent Consistency) uses optical flow to establish explicit motion correspondences between adjacent frames, and leverages these correspondences for both latent consistency constraints and cross-frame attention bias, reducing flickering and local drift in the generated video. Third, PMTR (Progressive Multi-Task Routing) coordinates the optimization rhythm of different tasks through conditional routing and curriculum-based task sampling, enabling the model to preserve task specialization while sharing a common backbone.

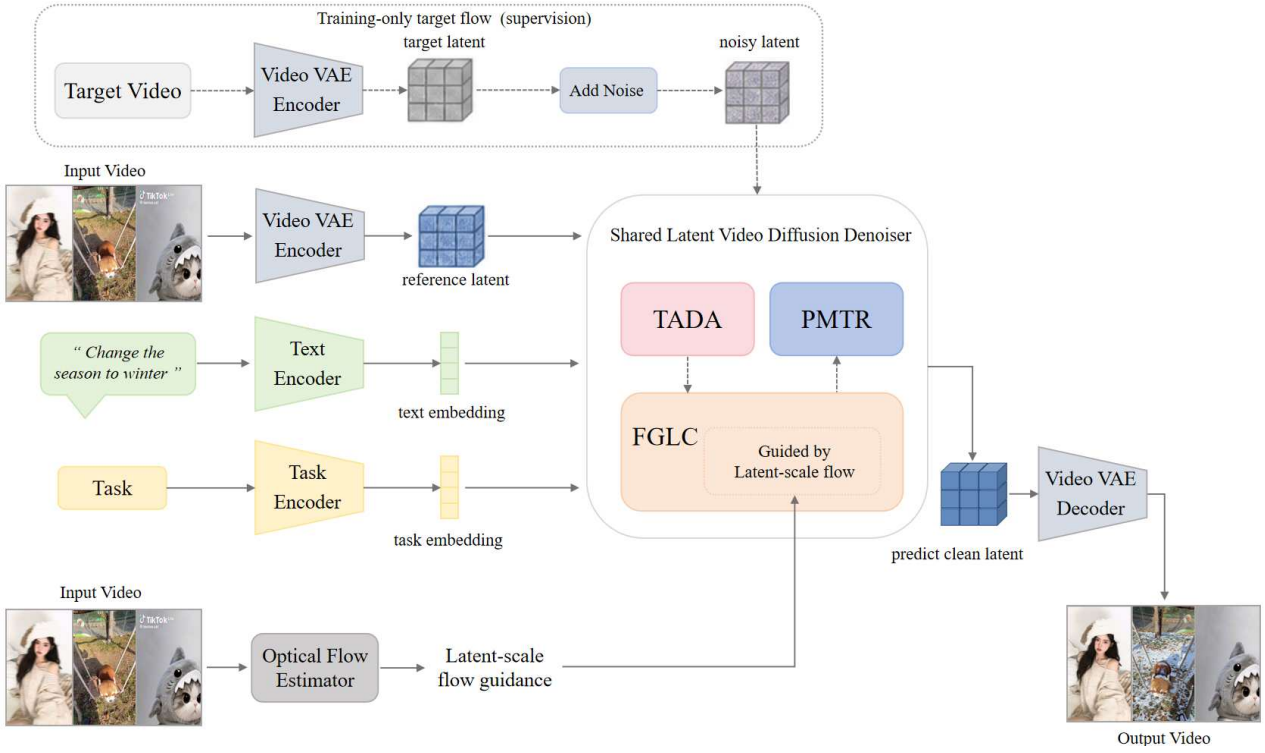


Figure 1: Overall architecture of **LTX-Insight**.

From the training perspective, we adopt a staged optimization strategy from easy to hard. Stage A optimizes only the core diffusion loss so that the model first acquires basic video generation and reconstruction capabilities. Stage B activates TADA to strengthen structure preservation and target-region modification under task conditions. Stage C introduces FGLC to extend single-frame quality constraints to cross-frame temporal consistency. Stage D enables PMTR routing and dynamic weighting, allowing the three tasks to jointly converge toward a unified objective. This pipeline follows the optimization logic of "first establish reliable reconstruction, then introduce temporal constraints, and finally perform multi-task collaboration," which helps reduce gradient oscillation caused by early-stage multi-objective competition.

3.2 Preliminaries: Instruction-Guided Latent Video Diffusion

Let the input video be $X \in \mathbb{R}^{T\times H_{\text{in}}\times W_{\text{in}}\times 3}$, and the training target video be $Y^* \in \mathbb{R}^{T\times H\times W\times 3}$. We first use a video VAE encoder $\mathcal{E}_{\text{vae}}(\cdot)$ to map the target video and the input video in each training sample into a compressed latent space:

$$z_0 = \mathcal{E}_{\text{vae}}(Y^*), \quad z_{\text{ref}} = \mathcal{E}_{\text{vae}}(X). \tag{1}$$

Here, z_0 denotes the target video latent variable, and z_{ref} denotes the reference latent variable encoded from the input video. For all tasks, X is the original input video provided to the model: it may be a low-quality video for super-resolution, an occluded video for occlusion removal, or an unedited source video for semantic editing. By representing all three task inputs as reference latent variables, the model can receive textual and visual conditions within a unified diffusion framework.

Since the input and target resolutions may differ in the super-resolution task, we use a task-dependent spatial alignment operator $\mathcal{R}_r(\cdot)$ before any pixel-level comparison. For super-resolution, $\mathcal{R}_{\text{sr}}(\cdot)$ upsamples the low-resolution input to the target resolution using bicubic interpolation followed by VAE-compatible cropping or padding; for occlusion removal and semantic editing, $\mathcal{R}_r(\cdot)$ is the identity operator after standard resizing. The aligned input frame is denoted as $x_i^a = \mathcal{R}_r(x_i)$, ensuring that all pixel-level losses are computed on the same $H \times W$ grid.

During training, the diffusion forward process adds Gaussian noise to the target latent variable z_0. To avoid confusion with the video frame index i, we use τ to denote the diffusion timestep:

$$z_\tau = \sqrt{\bar{\alpha}_\tau} z_0 + \sqrt{1-\bar{\alpha}_\tau}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{2}$$

Here, $\bar{\alpha}_\tau$ is the cumulative coefficient in the noise sched-

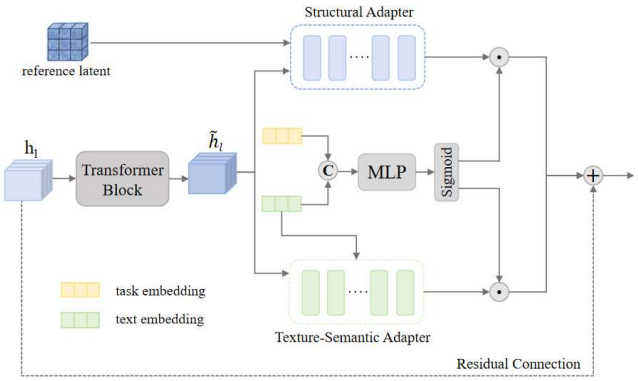


Figure 2: Overall architecture of **TADA**.

ule, and ϵ is standard Gaussian noise. The conditioning set is denoted by

$$c = \{E_{\text{text}}(p), E_{\text{task}}(r), z_{\text{ref}}\}, \tag{3}$$

where $E_{\text{text}}(\cdot)$ denotes the text encoder and $E_{\text{task}}(\cdot)$ denotes the task embedding layer. The basic diffusion loss is defined as

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{z_0,\tau,\epsilon,c}\left[\|\epsilon - \epsilon_\theta(z_\tau, \tau, c)\|_2^2\right]. \tag{4}$$

This loss enables the denoising network ϵ_θ to recover the target latent variable conditioned on the reference video, text instruction, and task label. It should be emphasized that the above latent-space noise prediction belongs to the standard video diffusion training paradigm. The key modifications of this work lie in the branch-separated task injection and weak decoupling constraint of TADA, the explicit temporal constraints of FGLC in latent space, and the multi-task routing and curriculum optimization of PMTR.

3.3 TADA: Task-Aware Branch-Separated Adaptation

The central difficulty of multi-task video post-processing is that different tasks demand different internal layer behaviors from the model. Super-resolution depends more on low- and mid-level features such as edges, textures, and frequency details. Occlusion removal requires semantic completion within local occluded regions while preserving continuity in the surrounding background. Semantic editing relies more heavily on high-level semantic representations and interaction with textual conditions. If all tasks share the same injection strength, the model tends to exhibit two types of failure: restoration tasks may be disturbed by the text-editing branch, causing unnecessary background rewriting, while semantic editing tasks may be over-constrained by reconstruction objectives, resulting in insufficient editing amplitude. The design motivation of TADA is therefore not to assume a strictly orthogonal separation between structure and semantics, but to weakly factorize their dominant adaptation directions under task-aware gates.

Formally, let h_l denote the hidden state of the l-th layer of the video diffusion Transformer, and let the task and text embeddings be $e_r = E_{\text{task}}(r)$ and $e_p = E_{\text{text}}(p)$, respectively. TADA first generates layer-wise gates according to the task label and text instruction:

$$[g_l^{\text{str}}, g_l^{\text{tex}}] = \sigma(W_l[e_r; e_p] + b_l)\,. \tag{5}$$

Here, $[\cdot;\cdot]$ denotes vector concatenation, $\sigma(\cdot)$ is the Sigmoid function, and g_l^{str} and g_l^{tex} control the injection strength of the structural branch and the texture-semantic branch at layer l, respectively. This gating mechanism allows the model to automatically adjust the adaptation magnitude at different layers for different tasks. For example, super-resolution usually emphasizes shallow-layer structural and texture branches, whereas semantic editing relies more on deep semantic branches.

In the layer update, TADA uses two branch-separated residual adapters, namely the structural branch $\mathcal{A}_l^{\text{str}}$ and the texture-semantic branch $\mathcal{A}_l^{\text{tex}}$:

$$h_l' = h_l + g_l^{\text{str}} \odot \mathcal{A}_l^{\text{str}}(h_l, \mathcal{P}_l(z_{\text{ref}})) + g_l^{\text{tex}} \odot \mathcal{A}_l^{\text{tex}}(h_l, e_p). \tag{6}$$

Here, $\odot$ denotes element-wise modulation along the channel or token dimension. $\mathcal{P}_l(\cdot)$ is a layer-wise latent alignment projection that maps z_{ref} to the spatial token grid and channel dimension of h_l. In the super-resolution task, this projection includes latent-grid interpolation and a lightweight linear projection, while in occlusion removal and semantic editing it mainly performs token reshaping and channel projection. The structural branch takes the aligned reference latent variable $\mathcal{P}_l(z_{\text{ref}})$ as its primary condition and is responsible for constraining geometric layout, subject contours, and consistency of non-target regions. The texture-semantic branch takes the text embedding e_p as its primary condition and is used to introduce task-related texture enhancement, occlusion completion, or semantic changes. Both branches can be implemented as lightweight MLP adapters, and are equipped with zero-initialized residual connections so as not to damage the generation capability of the pretrained video diffusion backbone at the beginning of training. Since structure and semantics are not strictly independent in real editing cases, the two branches are treated as weakly decoupled pathways rather than fully disentangled latent factors.

In our setting, TADA does not rely on explicit occlusion or editing masks. Instead, task-specific behav-

ior is learned from paired supervision and instruction-conditioned routing. The paired reconstruction term is defined as

$$\mathcal{L}_{\text{pair}} = \frac{1}{T}\sum_{i=1}^{T} \|\hat{y}_i - y_i^*\|_1 . \tag{7}$$

Here, $\hat{y}_i$ is the i-th output frame and y_i^* is the i-th target frame. For super-resolution and occlusion removal, y_i^* is the clean target frame; for semantic editing, y_i^* is the curated pseudo-reference frame. Non-target preservation is therefore encouraged by the paired target construction itself rather than by an explicit mask: the target video keeps the content that should remain unchanged and modifies only the regions required by the instruction.

To make the branch separation more than a purely architectural assumption, we add a weak orthogonality regularizer on the gated adapter responses. Let R_l^{str} and R_l^{tex} denote the token-by-channel matrices obtained by reshaping $g_l^{\text{str}} \odot \mathcal{A}_l^{\text{str}}(h_l, \mathcal{P}_l(z_{\text{ref}}))$ and $g_l^{\text{tex}} \odot \mathcal{A}_l^{\text{tex}}(h_l, e_p)$, respectively. After token-wise ℓ_2 normalization, the regularizer is defined as

$$\mathcal{L}_{\text{orth}} = \frac{1}{L}\sum_{l=1}^{L} \left\| \bar{R}_l^{\text{str}\top} \bar{R}_l^{\text{tex}} \right\|_F^2 . \tag{8}$$

This term discourages the two adapters from learning highly overlapping response directions while still allowing them to interact through the shared hidden state. The final TADA objective is therefore

$$\mathcal{L}_{\text{TADA}} = \mathcal{L}_{\text{pair}} + \lambda_{\text{orth}}\mathcal{L}_{\text{orth}}. \tag{9}$$

Together with the gated adapters, this weakly decoupled objective allows the model to perform the required task-specific modifications while reducing unnecessary changes outside the instruction-relevant content.

3.4 FGLC: Flow-Guided Latent Temporal Consistency

Relying only on frame-wise reconstruction losses is insufficient to ensure stable playback, because the same object or texture position changes across neighboring frames due to motion. If one directly constrains the pixel difference between $\hat{y}_i$ and $\hat{y}_{i+1}$, real motion may be undesirably suppressed. The design motivation of FGLC is to establish motion-compensated correspondences in latent space using optical flow, so that the model enforces consistency after motion alignment rather than static pixel invariance.

We first use a frozen optical flow estimator $\mathcal{F}(\cdot)$ to estimate motion fields between adjacent input frames:

$$F_{i\to i+1} = \mathcal{F}(x_i, x_{i+1}), \quad \bar{F}_{i\to i+1} =\downarrow F_{i\to i+1}. \tag{10}$$

Here, $\downarrow$ denotes spatial downsampling aligned with the VAE compression ratio, and $\bar{F}_{i\to i+1}$ is the optical flow at latent resolution. Based on this motion field, we warp the predicted latent variable of frame i into the coordinate system of frame $i+1$:

$$\tilde{z}_{i+1} = W\big(\hat{z}_i, \bar{F}_{i\to i+1}\big), \tag{11}$$

where $W(\cdot)$ is a differentiable warping operator, $\hat{z}_i$ denotes the predicted clean latent variable or intermediate denoised latent variable of frame i, and $\tilde{z}_{i+1}$ denotes the aligned prior propagated from the previous frame. This term provides explicit motion guidance for the subsequent consistency constraints.

Since occlusion, out-of-frame motion, and fast movement may lead to unreliable flow estimation, we construct a validity map using forward-backward consistency:

$$O_{i+1} = \mathbf{1}(\|F_{i\to i+1} + W(F_{i+1\to i}, F_{i\to i+1})\|_2 < \delta), \tag{12}$$

where O_{i+1} is the visibility map and δ is the forward-backward consistency threshold. Only regions satisfying this condition are treated as reliable correspondences. Based on this validity map, the flow-guided latent consistency loss is defined as

$$\mathcal{L}_{\text{flow}} = \frac{\sum_{i=1}^{T-1} \|O_{i+1} \odot (\hat{z}_{i+1} - \tilde{z}_{i+1})\|_1}{\sum_{i=1}^{T-1} \|O_{i+1}\|_1 + \varepsilon}. \tag{13}$$

The denominator term ε is used to avoid numerical instability when the valid region is empty. This loss is computed in latent space rather than pixel space, which reduces the computational overhead of high-resolution video training and avoids overly strong penalties from pixel-level texture differences.

In addition to explicit latent alignment, FGLC also injects motion priors into cross-frame attention. Let Q_i, K_j, and V_j denote the query of frame i and the key/value tokens of frame j, respectively. The flow-guided attention is written as

$$\text{Attn}_{i\to j} = \text{Softmax}\left(\frac{Q_i K_j^\top}{\sqrt{d}} + \eta B_{i\to j}\right) V_j. \tag{14}$$

Here, $B_{i\to j}$ is a spatial proximity bias constructed from $\bar{F}_{i\to j}$, which takes larger values when two tokens are closer after flow alignment; η is the bias strength; and d is the attention channel dimension. This design encourages cross-frame attention to preferentially aggregate motion-consistent tokens, thereby reducing random texture fluctuations across adjacent frames.

The final temporal objective is written as

$$\mathcal{L}_{\text{FGLC}} = \mathcal{L}_{\text{flow}} + \lambda_{\text{attn}} \frac{1}{T-1}\sum_{i=2}^{T} \left\| A^{(i)} - A^{(i-1)} \right\|_1 . \tag{15}$$

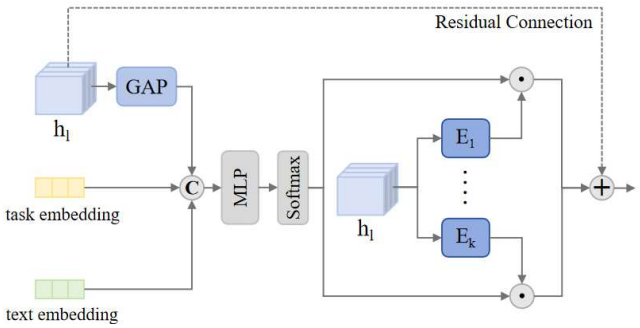


Figure 3: Overall architecture of **PMTR**.

Here, $A^{(i)}$ denotes the cross-frame attention distribution of the i-th frame. The first term enforces motion-compensated latent consistency, while the second term suppresses abrupt variations in the attention pattern across frames. The two are complementary: the former provides explicit geometric constraints, while the latter stabilizes texture and semantic propagation at the feature interaction level.

3.5 PMTR: Progressive Multi-Task Routing Collaboration

Directly mixing the three tasks within a unified model is easily affected by differences in sample scale, loss magnitude, and task difficulty. For example, super-resolution samples usually provide dense supervision and converge faster, while semantic editing samples depend more heavily on textual conditions and high-level semantics and therefore exhibit larger training fluctuation. If random sampling is performed without distinction, the model may be biased toward easier restoration tasks, resulting in insufficient editing capability. The design motivation of PMTR is to introduce lightweight expert routing on top of a shared backbone and gradually expand the task distribution through curriculum scheduling, thereby reducing negative transfer.

Assume that the l-th layer contains K lightweight expert branches $\{E_{l,k}\}_{k=1}^{K}$. The router computes expert activation probabilities from the current hidden state, task embedding, and text embedding:

$$u_{l,k} = w_{l,k}^{\top}[\mathrm{Pool}(h_l); e_r; e_p] + b_{l,k}, \quad \pi_{l,k} = \mathrm{Softmax}_k(u_{l,k}). \tag{16}$$

Here, $\mathrm{Pool}(\cdot)$ denotes mean pooling over the token dimension, $u_{l,k}$ is the routing logit of the k-th expert, and $\pi_{l,k}$ is the corresponding activation probability. The routed hidden state is then given by

$$h_l^{\mathrm{route}} = h_l + \sum_{k=1}^{K} \pi_{l,k} E_{l,k}(h_l). \tag{17}$$

This form allows different tasks to share the backbone parameters while activating different expert subspaces at local layers. Compared with maintaining an independent model for each task, PMTR has lower parameter and deployment cost; compared with a fully shared model, it preserves task-specialized representations.

At the sampling level, PMTR uses curriculum scheduling to control the task distribution during training:

$$\begin{aligned} \gamma(s) &= \min\Big(1, \frac{s}{S}\Big), \\ p_s(r) &= (1-\gamma(s))p_{\mathrm{easy}}(r) \\ &\quad + \gamma(s)p_{\mathrm{full}}(r). \end{aligned} \tag{18}$$

Here, s is the current training step, S is the curriculum transition length, $p_{\mathrm{easy}}(r)$ denotes an early-stage sampling distribution biased toward restoration tasks, and $p_{\mathrm{full}}(r)$ denotes the target distribution covering all tasks. As $\gamma(s)$ increases from 0 to 1, the training data gradually shifts from relatively stable restoration tasks to the full task set including semantic editing, thus preventing the model from being disturbed too early by complex editing tasks before it forms reliable spatio-temporal representations.

To handle the inconsistency of loss scales across tasks, we adopt an uncertainty-weighted multi-task objective and further introduce expert load balancing and entropy regularization:

$$\mathcal{L}_{\mathrm{PMTR}} = \mathcal{L}_{\mathrm{unc}} + \lambda_{\mathrm{bal}}\mathcal{L}_{\mathrm{bal}} - \lambda_{\mathrm{ent}}\mathcal{L}_{\mathrm{ent}}. \tag{19}$$

$$\mathcal{L}_{\mathrm{unc}} = \sum_{r} \left(\frac{1}{2\sigma_r^2}\mathcal{L}_r + \log \sigma_r \right), \tag{20}$$

$$\mathcal{L}_{\mathrm{bal}} = \sum_{k=1}^{K} \left(\bar{\pi}_k - \frac{1}{K} \right)^2, \quad \mathcal{L}_{\mathrm{ent}} = \sum_{l} \mathcal{H}(\pi_l). \tag{21}$$

Here, $\mathcal{L}_r$ denotes the supervision loss associated with task r, and σ_r is a learnable task uncertainty parameter; when a task is noisier or harder to optimize, σ_r can adaptively adjust its effective weight. $\bar{\pi}_k$ denotes the average load of expert k within a batch; the load balancing term prevents a few experts from being overused. $\mathcal{H}(\pi_l)$ denotes the entropy of the routing distribution at layer l; entropy regularization is used to avoid early-stage routing collapse to a single expert. Through these designs, PMTR simultaneously addresses task-weight imbalance and expert-utilization imbalance.

3.6 Overall Objective and Stage-Wise Weight Scheduling

The overall training objective is

$$\begin{aligned} \mathcal{L}_{\mathrm{total}}(s) &= \mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{tada}}(s)\mathcal{L}_{\mathrm{TADA}} \\ &\quad + \lambda_{\mathrm{fglc}}(s)\mathcal{L}_{\mathrm{FGLC}} \\ &\quad + \lambda_{\mathrm{pmtr}}(s)\mathcal{L}_{\mathrm{PMTR}}. \end{aligned} \tag{22}$$

Here, $\mathcal{L}_{\text{diff}}$ is the basic denoising objective, ensuring alignment between the generated distribution and the target video latent variables; $\mathcal{L}_{\text{TADA}}$ is responsible for target-region editing and non-target-region preservation; $\mathcal{L}_{\text{FGLC}}$ enforces cross-frame motion-compensated consistency; and $\mathcal{L}_{\text{PMTR}}$ handles multi-task weighting, expert load balancing, and routing stability. The coefficients $\lambda_{\text{tada}}(s)$, $\lambda_{\text{fglc}}(s)$, and $\lambda_{\text{pmtr}}(s)$ are module weights that vary dynamically with the training step.

The weights are activated using a smooth scheduling function:

$$\lambda_m(s) = \frac{\lambda_m^{\max}}{1 + \exp[-a_m(s - S_m)]}, \quad m \in \{\text{tada}, \text{fglc}, \text{pmtr}\}. \tag{23}$$

Here, $\lambda_m^{\max}$ is the maximum weight of module m, a_m controls the growth slope, and S_m is the activation center step. In practice, we set $S_{\text{tada}} < S_{\text{fglc}} < S_{\text{pmtr}}$, i.e., the model first learns task-aware injection, then temporal constraints, and finally full multi-task routing collaboration. Compared with activating all losses simultaneously from the very beginning of training, this stage-wise schedule reduces multi-objective gradient conflicts and mitigates early-stage problems such as over-suppressed editing amplitude or unstable temporal constraints, thereby improving the convergence stability of unified video restoration and semantic editing.

4 Experiments

4.1 Dataset and Task Construction

We construct a multi-task video post-processing dataset $\mathcal{D} = \mathcal{D}_{\text{sr}} \cup \mathcal{D}_{\text{occ}} \cup \mathcal{D}_{\text{edit}}$, where the three subsets correspond to super-resolution, occlusion removal, and semantic editing, respectively. All source videos are collected from publicly accessible online video sources, including general video-sharing and short-video platforms, and are used only for research analysis. The dataset contains 2,000 videos in total, including 600 videos for super-resolution, 600 videos for occlusion removal, and 800 videos for semantic editing. Each sample is uniformly represented as (X, p, r, Y^*), where X is the input video, p is the text instruction or task description, r is the task label, and Y^* is the supervision target video. The original online videos and generated target videos are not redistributed, and personally identifiable information is removed or anonymized when necessary.

The super-resolution subset $\mathcal{D}_{\text{sr}}$ contains 600 videos and is generated by degrading high-quality videos through a degradation model. The degradation process includes bicubic downsampling, blur-kernel perturbation, compression noise, and mild sensor noise to simulate multi-source degradations in real low-quality videos. The training target is the original high-quality video, and the instruction p uses a unified template such as "increase the resolution and restore details." This setup is used to evaluate detail recovery and frequency-domain reconstruction stability.

The occlusion removal subset $\mathcal{D}_{\text{occ}}$ contains 600 videos and is constructed from clean source videos without visible target occlusions. We use the clean videos as the supervision targets Y^*, and synthesize the corresponding occluded inputs X by adding watermark-like overlays, subtitles, logos, text labels, and temporary markers. These overlays are synthetic visual obstructions used for restoration research rather than copyright, authorship, or ownership marks. The occlusion types cover both text-like and non-text occlusions. During synthesis, the occlusion position, transparency, scale, color, shape, duration, and motion trajectory are randomly sampled. No occlusion mask is used as model input; the model receives only the occluded video, the task label, and the removal prompt.

The semantic editing subset $\mathcal{D}_{\text{edit}}$ contains 800 videos and is constructed through a model-assisted curation pipeline. For the training and validation splits, we first generate candidate edited videos from the online source videos using multiple commercial closed-source video editing models under manually designed natural-language instructions. The generated candidates are then manually screened according to instruction consistency, preservation of non-target regions, temporal stability, and artifact level. Only candidates that satisfy these quality criteria are retained as pseudo-supervised editing pairs. The editing scenarios cover object attribute modification, local replacement, mild background adjustment, and stylized enhancement.

To reduce generator-specific bias in evaluation, the semantic editing test split is constructed using three different commercial closed-source models that are not used for generating the training and validation samples. The test set is further stratified by editing type, motion magnitude, and scene complexity. All data are split at the video level into training, validation, and test sets with a ratio of 70%/10%/20%, ensuring that different clips from the same original video do not appear in different splits, thus avoiding data leakage. Specifically, the super-resolution subset is divided into 420/60/120 videos, the occlusion removal subset into 420/60/120 videos, and the semantic editing subset into 560/80/160 videos for training, validation, and testing, respectively. Since the semantic editing targets are generated and manually curated rather than captured as human-created ground truth, we treat them as high-quality pseudo references throughout training and evaluation.

In the unified preprocessing stage, videos are cropped or padded to match the frame-count format supported

by the model, such that the number of frames satisfies $T = 8k + 1$; the spatial resolution is adjusted to sizes compatible with the VAE and Transformer. Before evaluation, all videos are processed with the same color space, frame rate, and normalization strategy to ensure fair comparison across different methods.

4.2 Implementation Details

LTX-Insight is built upon a unified video diffusion backbone, where the video VAE, text encoder, and spatio-temporal Transformer are shared across the three tasks. TADA, FGLC, and PMTR are inserted into the backbone as lightweight trainable modules. Except for the adapters, routers, and necessary normalization parameters, the remaining pretrained parameters can either be frozen or fine-tuned with a low learning rate depending on the experimental setup. We use bfloat16 mixed-precision training to reduce memory consumption, and adopt AdamW together with cosine learning-rate decay and linear warm-up. The default implementation settings are summarized in Table 1.

Table 1: Default implementation settings used in the experiments.

Item	**Setting**
Backbone	Latent video diffusion Transformer
Input clip length	49 frames
Training resolution	512×512 with temporally consistent random cropping
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.999$
Learning rate	1.0×10^{-5}
Weight decay	0.01
Precision	bfloat16 mixed precision
Hardware	8 NVIDIA H800 GPUs

During training, the task sampling distribution gradually transitions from restoration tasks to the complete multi-task distribution according to the curriculum function $\gamma(s)$ defined in Section 3.5. Optical flow is precomputed and cached offline using a frozen estimator to reduce repeated computation during training. For FGLC, only regions that pass the forward-backward consistency check participate in the latent temporal loss. For PMTR, expert load is measured at the batch level, and routing entropy as well as expert utilization are monitored after each training epoch.

During inference, we adopt a single-model, single-entry setting by default, where the output is directly generated from the input video, task label, and text instruction. All methods are run under the same hardware, the same input resolution, the same sampling steps, and the same set of random seeds.

4.3 Evaluation Metrics

We select evaluation metrics according to the focus of each task. For super-resolution, we report PSNR, SSIM, and tLPIPS to jointly evaluate pixel fidelity, structural similarity, and cross-frame stability. For occlusion removal, we report PSNR, SSIM, and LPIPS to assess reconstruction quality, structural preservation, and perceptual naturalness. For semantic editing, we report CLIP Score and FID to measure instruction following and distribution quality, and additionally include User Vote in the external comparison to evaluate subjective preference. Let the model output video be $\hat{Y} = \{\hat{y}_t\}_{t=1}^{T}$ and the reference video be $Y = \{y_t\}_{t=1}^{T}$. For semantic editing, Y denotes the curated pseudo reference generated by external commercial models rather than an absolute human-created ground truth. Except for user voting, all metrics are first averaged within each video and then aggregated over the test set.

4.3.1 PSNR and SSIM

PSNR measures the pixel-level reconstruction error between the output video and the reference video. A higher value indicates better reconstruction fidelity. It is defined as

$$\mathrm{MSE} = \frac{1}{TCHW} \sum_{t=1}^{T} \|\hat{y}_t - y_t\|_2^2 \tag{24}$$

$$\mathrm{PSNR} = 10 \log_{10} \frac{\mathrm{MAX}^2}{\mathrm{MSE}}. \tag{25}$$

Here, C, H, and W denote the number of channels, height, and width, respectively, and MAX is the maximum pixel value. SSIM [18] measures structural similarity between two frames from the aspects of luminance, contrast, and structure. Video-level SSIM is computed by averaging the frame-wise SSIM:

$$\mathrm{SSIM} = \frac{1}{T} \sum_{t=1}^{T} \mathrm{SSIM}(\hat{y}_t, y_t). \tag{26}$$

Both PSNR and SSIM are the higher-the-better metrics. We report them for the super-resolution and occlusion removal tasks to evaluate reconstruction accuracy and structural preservation.

4.3.2 LPIPS and tLPIPS

LPIPS [19] measures the difference between the output video and the reference video in a deep perceptual feature space. Lower values indicate that the perceptual quality is closer to the reference video:

$$\mathrm{LPIPS} = \frac{1}{T} \sum_{t=1}^{T} \sum_{l} \frac{1}{H_l W_l} \left\| w_l \odot (\phi_l(\hat{y}_t) - \phi_l(y_t)) \right\|_2^2. \tag{27}$$

Here, $\phi_l(\cdot)$ denotes the feature map of the l-th layer in the perceptual network, and w_l is the channel weight. We report LPIPS for the occlusion removal task to measure the perceptual naturalness of the completed video after occlusion repair.

tLPIPS evaluates motion-compensated perceptual consistency across adjacent frames. Lower values indicate less cross-frame flickering and texture jitter:

$$\text{tLPIPS} = \frac{1}{T-1}\sum_{t=1}^{T-1}\text{LPIPS}\left(\hat{y}_{t+1}, W(\hat{y}_t, F_{t\to t+1})\right). \tag{28}$$

Here, $F_{t\to t+1}$ is the optical flow between adjacent frames, and $W(\cdot)$ is the warping operator. We report tLPIPS for the super-resolution task to evaluate the temporal stability of the enhanced details during playback.

4.3.3 CLIP Score

CLIP Score is based on CLIP [20] and measures the consistency between the semantic editing result and the text instruction. Given a text instruction p and an output video $\hat{Y}$, we first extract the image feature of each frame and average them, then compute cosine similarity with the text feature:

$$\text{CLIP Score} = \cos\left(f_{\text{text}}(p), \frac{1}{T}\sum_{t=1}^{T} f_{\text{img}}(\hat{y}_t)\right). \tag{29}$$

Here, $f_{\text{text}}(\cdot)$ and $f_{\text{img}}(\cdot)$ denote the text encoder and image encoder of CLIP, respectively. A higher CLIP Score indicates that the output video better follows the editing instruction.

4.3.4 FID

FID [21] measures the discrepancy between the distribution of generated video frames and that of real reference frames. We feed the video frames into a feature extraction network to obtain the feature distributions of the generated results (μ_g, Σ_g) and the real references (μ_r, Σ_r), respectively. The metric is defined as

$$\text{FID} = \|\mu_g - \mu_r\|_2^2 + \text{Tr}\left(\Sigma_g + \Sigma_r - 2(\Sigma_g\Sigma_r)^{1/2}\right). \tag{30}$$

Lower FID indicates that the generated results are closer to real data in the overall visual distribution. We report FID for the semantic editing task to evaluate the naturalness and distribution quality of edited videos.

4.3.5 User Vote

User Vote is used to complement automatic metrics when they are insufficient to characterize subjective quality. We adopt a pairwise preference evaluation protocol with 50 video pairs and 5 independent evaluators. For each comparison, the results of LTX-Insight and a competing method are randomly shuffled and shown side by side. Evaluators are asked to choose the better result according to task completion, temporal stability, preservation of non-target regions, and overall visual quality. The final User Vote denotes the proportion of pairwise comparisons in which a method is preferred. A higher value indicates better subjective quality. This metric is mainly used for the semantic editing task, because the success of semantic editing depends not only on text consistency, but also on whether the edited region looks natural, whether the non-edited region remains stable, and whether noticeable flickering occurs during continuous playback.

4.4 Ablation Studies

4.4.1 Super-Resolution Task

Table 2: Ablation study on the super-resolution task.

Method	PSNR ↑	SSIM ↑	tLPIPS ↓
Baseline	37.84	0.9441	4.63
+ PMTR	38.02	0.9463	4.31
+ TADA	38.27	0.9496	4.47
+ TADA + FGLC	38.42	0.9510	3.87
Full Model	**38.53**	**0.9522**	**3.75**

As shown in Table 2, the super-resolution results show that adding PMTR alone yields a modest improvement in PSNR and SSIM over the Baseline, indicating that progressive task routing can improve the stability of shared representations in multi-task training. After adding TADA, PSNR improves from 37.84 to 38.27 and SSIM improves from 0.9441 to 0.9496, suggesting that task-aware branch-separated adaptation enhances structure preservation and detail recovery. With the further introduction of FGLC, tLPIPS decreases from 4.47 to 3.87, indicating that flow-guided latent consistency effectively reduces cross-frame texture jitter. The full model achieves the best performance on all three metrics, validating the complementarity of TADA, FGLC, and PMTR in the super-resolution scenario.

4.4.2 Occlusion Removal Task

As shown in Table 3, occlusion removal imposes higher requirements on local completion and background consistency, and therefore the effect of different modules is even more pronounced. Compared with the Baseline,

Table 3: Ablation study on the occlusion removal task.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
Baseline	20.12	0.7463	0.338
+ PMTR	22.35	0.7757	0.249
+ TADA	22.78	0.8152	0.276
+ TADA + FGLC	24.12	0.8394	0.214
Full Model	**24.45**	**0.8479**	**0.182**

TADA improves PSNR by 2.66 dB and substantially reduces LPIPS, indicating that the weakly decoupled structural and texture-semantic branches can reduce background contamination during occluded-region repair. FGLC further improves SSIM and lowers LPIPS, suggesting that motion-compensation constraints help alleviate inconsistency of repaired content across neighboring frames. Compared with the Baseline, the full model achieves clear improvements in PSNR, SSIM, and LPIPS, indicating that multi-task routing can further strengthen the specialized representation of occlusion removal while maintaining overall training stability.

4.4.3 Semantic Editing Task

Table 4: Ablation study on the semantic editing task.

Method	CLIP Score ↑	FID ↓
Baseline	0.902	32.74
+ PMTR	0.919	31.86
+ TADA	0.910	29.12
+ TADA + FGLC	0.925	27.87
Full Model	**0.937**	**27.13**

As shown in Table 4, in the semantic editing task, the Baseline already exhibits a certain degree of text responsiveness, but its CLIP Score and FID are still affected by semantic interference under mixed-task training. After adding TADA, FID decreases from 32.74 to 29.12, indicating that task-aware gating improves the visual distribution quality of edited videos. With the further introduction of FGLC, the CLIP Score rises from 0.910 to 0.925 and FID decreases from 29.12 to 27.87, suggesting that temporal consistency constraints not only reduce flicker but also help stabilize semantic editing results. The full model achieves the highest CLIP Score of 0.937 and the lowest FID of 27.13, indicating that PMTR can alleviate negative transfer among tasks in complex editing scenarios and better balance semantic modification and visual quality.

4.4.4 Fine-Grained Ablation of TADA

To further analyze the contribution of the two branch-separated pathways in TADA, we conduct a compact component-level ablation on occlusion removal and semantic editing. The complete variant keeps both the structural branch and the texture-semantic branch active, while the other two variants remove either the structural branch or the texture-semantic branch.

As shown in Table 6, removing the structural branch clearly degrades occlusion removal, especially in PSNR and LPIPS, because the model loses an explicit pathway for preserving geometric layout and surrounding-region consistency. Removing the texture-semantic branch has a stronger impact on semantic editing, where the CLIP Score drops from 0.937 to 0.916 and FID increases from 27.13 to 30.18. These results indicate that the two branches play complementary roles: the structural branch mainly supports faithful reconstruction and background continuity, while the texture-semantic branch is more important for instruction-aligned appearance modification.

4.5 External Comparison Results

To further verify the effectiveness of LTX-Insight relative to existing methods, we perform external comparison experiments on the three tasks separately. Since different tasks have different objectives and evaluation metrics, we do not mix all methods into a single table, but instead report the results for super-resolution, occlusion removal, and semantic editing independently.

4.5.1 Super-Resolution Comparison

The super-resolution task mainly compares the performance of different models in terms of detail recovery, perceptual quality, and cross-frame stability. The compared methods include the classic recurrent super-resolution baseline BasicVSR [22], the unified video restoration framework VRT [4], and the diffusion-based temporally consistent super-resolution method StableVSR [23]. All methods are evaluated under the same low-quality inputs, the same test split, and the same evaluation scripts.

As shown in Table 7, compared with video super-resolution baselines, LTX-Insight achieves the best PSNR and SSIM, indicating that the unified diffusion backbone still maintains strong reconstruction capability after introducing task-aware adaptation. Compared with StableVSR, LTX-Insight further reduces tLPIPS, showing that the proposed FGLC contributes positively to temporal texture stability. These results suggest that under unified multi-task modeling, LTX-Insight does not sacrifice the performance of the super-resolution

Table 5: External comparison on the semantic editing task.

Method	CLIP Score ↑	FID ↓	User Vote ↑
T2V-Zero [8]	0.914	30.24	14.6%
TAV [5]	0.928	27.33	17.8%
Rerender A Video [13]	**0.945**	**26.95**	**37.4%**
LTX-Insight	0.937	27.13	30.2%

Table 6: Fine-grained ablation of TADA components.

Variant	Occ. PSNR ↑	Occ. SSIM ↑	Occ. LPIPS ↓	Edit CLIP ↑	Edit FID ↓
w/o structural branch	23.61	0.8418	0.204	0.921	29.04
w/o texture-semantic branch	23.93	0.8427	0.195	0.916	30.18
Full TADA	**24.45**	**0.8479**	**0.182**	**0.937**	**27.13**

Table 7: External comparison on the super-resolution task.

Method	PSNR ↑	SSIM ↑	tLPIPS ↓
BasicVSR [22]	37.57	0.9394	4.81
VRT [4]	37.69	0.9446	4.39
StableVSR [23]	38.12	0.9495	3.97
LTX-Insight	**38.53**	**0.9522**	**3.75**

task, but instead obtains more stable video outputs through latent temporal constraints.

4.5.2 Occlusion Removal Comparison

The occlusion removal task focuses on the quality of occluded-region reconstruction, overall perceptual quality, and structural fidelity. The compared methods include the propagation-enhanced video inpainting method ProPainter [24], the end-to-end flow-guided video inpainting method E2FGVI [25], and the diffusion-based video inpainting method DiffuEraser [26]. We mainly report PSNR, SSIM, and LPIPS, which evaluate different methods from the perspectives of pixel error, structural preservation, and perceptual naturalness.

Table 8: External comparison on the occlusion removal task.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
ProPainter [24]	21.26	0.7252	0.307
E2FGVI [25]	24.18	0.8454	0.194
DiffuEraser [26]	21.94	0.7538	0.432
LTX-Insight	**24.45**	**0.8479**	**0.182**

As shown in Table 8, in the occlusion removal task, LTX-Insight outperforms existing restoration methods on PSNR, SSIM, and LPIPS. This observation suggests that when only frame-wise restoration or a single inpainting model is used, occluded-region completion is prone to texture discontinuity and cross-frame drift. In contrast, LTX-Insight jointly constrains the output through reference latent variables, task conditions, and flow-guided consistency, enabling the occluded region to blend more naturally with the surrounding background. In particular, the lower LPIPS indicates that the model improves not only pixel-level fidelity but also perceptual naturalness.

4.5.3 Semantic Editing Comparison

The semantic editing task jointly evaluates instruction following, preservation of non-target regions, and cross-frame stability. The compared methods include Text2Video-Zero [8], Tune-A-Video [5], and Rerender A Video [13].

As reported in Table 5, semantic editing results show that LTX-Insight outperforms T2V-Zero and Tune-A-Video in CLIP Score, FID, and User Vote, while achieving competitive performance compared with Rerender A Video. Compared with T2V-Zero and Tune-A-Video, the higher CLIP Score indicates that the model responds more accurately to text instructions, while the lower FID suggests that the edited videos are more visually natural in the overall distribution. Although Rerender A Video obtains the best results in this comparison, LTX-Insight remains close to it while using a unified multi-task framework. These results validate the necessity of jointly modeling task labels, textual conditions, and visual reference conditions.

4.6 Visualization Analysis

We further provide qualitative results for occlusion removal, super-resolution, and semantic editing. The visualizations focus on local reconstruction quality, temporal stability, and the preservation of non-target regions.

As shown in Figure 4, LTX-Insight removes text-like occlusions while keeping the repaired regions visually coherent with the surrounding content. The completed regions remain stable across frames, suggesting that FGLC helps suppress temporal artifacts during dynamic occlusion removal.

Figure 5 shows that LTX-Insight restores clearer facial contours and fine textures from low-resolution inputs without obvious over-sharpening. This is consistent with the quantitative gains in PSNR, SSIM, and tLPIPS.

For semantic editing, Figure 6 demonstrates that the model can modify scene attributes while preserving the subject and main spatial structure. The edited appearance remains consistent across adjacent frames, indicating that TADA and FGLC jointly improve controllability and temporal coherence.



Figure 4: Visualization of text-like occlusion removal.



Figure 5: Visualization of detail recovery for super-resolution.

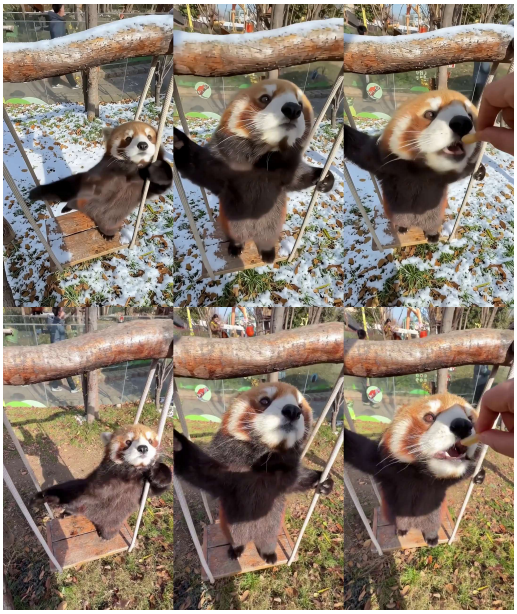


Figure 6: Visualization of scene-attribute modification for semantic editing.

5 Conclusion

In this paper, we propose LTX-Insight, a unified framework for video restoration and semantic editing, which achieves collaborative multi-task modeling on a single video diffusion backbone for video super-resolution, general occlusion removal, and instruction-driven se-

mantic editing. To address the common challenges in unified modeling, including task interference, cross-frame instability, and difficulties in joint training, we further design three complementary modules: TADA improves controllable injection across different tasks through task-aware branch-separated adaptation with weak orthogonality regularization; FGLC enhances temporal stability through flow-guided latent consistency constraints; and PMTR mitigates negative transfer and improves convergence stability through progressive task routing and curriculum learning.

Experimental results show that LTX-Insight consistently delivers effective improvements across all three tasks: it balances detail restoration and cross-frame consistency in super-resolution, performs more natural local completion while preserving background continuity in occlusion removal, and better balances instruction following, preservation of non-target regions, and overall visual quality in semantic editing. The ablation studies further validate the effectiveness and complementarity of TADA, FGLC, and PMTR, indicating that weak branch separation, temporal constraints, and routing collaboration are all crucial for improving the final video quality under a unified framework.

Overall, this work provides a scalable technical path for single-model, multi-task, controllable video post-processing, and lays a foundation for building more general video editing systems in the future. Future work may focus on training with larger-scale real occlusion data and modeling longer video sequences, so as to further improve the generalization and practical value of unified video editing frameworks in complex real-world scenarios. In addition, the current model is still less robust when removing moving semi-transparent overlays, especially when the overlay has strong transparency changes, large motion, or overlaps with complex foreground objects. Its super-resolution performance also remains limited for severely degraded old videos, where strong compression artifacts, motion blur, film grain, and missing high-frequency details make faithful restoration difficult. Improving dynamic overlay localization, temporally stable restoration, and restoration under severe real-world degradation will be important directions for future work.

Data and Ethical Considerations

All source videos used in this study are collected from publicly accessible online sources and are used only for research analysis. We do not redistribute the original online videos, the generated editing targets, or the outputs of external commercial models. When necessary, personally identifiable information is removed or anonymized, and qualitative examples are selected to avoid exposing sensitive identity information.

The occlusion removal task in this paper is designed for restoration under synthetic or research-controlled visual obstructions. The watermark-like overlays used in our dataset are artificially generated occlusions and should not be interpreted as encouraging the removal of copyright notices, ownership marks, creator attribution, platform logos, or other legally protected identifiers from third-party content. The proposed method is intended for legitimate restoration, editing, and post-processing scenarios where users have appropriate rights or permission to process the input videos.

For semantic editing, target videos are generated with external commercial closed-source models and then manually screened as pseudo references. These generated targets are used for training and evaluation only, are not released as a public dataset, and are described as pseudo references rather than human-created ground truth. Users of similar pipelines should follow the terms of service of the corresponding commercial systems and respect the rights of content owners.

Competing Interests

Both authors are affiliated with Chongqing Valiant Cat Technology Co., Ltd., which organized this research. The authors declare that they have no additional competing financial or non-financial interests.

References

[1] U. Singer et al. Make-A-Video: Text-to-Video Generation without Text-Video Data. In *Proc. ICLR*, 2023.

[2] J. Ho et al. Imagen Video: High Definition Video Generation with Diffusion Models. *arXiv:2210.02303*, 2022.

[3] A. Blattmann et al. Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. In *Proc. CVPR*, 2023.

[4] J. Liang et al. VRT: A Video Restoration Transformer. *arXiv:2201.12288*, 2022.

[5] J. Z. Wu et al. Tune-A-Video: One-Shot Tuning of Image Diffusion Models for Text-to-Video Generation. In *Proc. ICCV*, 2023.

[6] C. Qi et al. FateZero: Fusing Attentions for Zero-shot Text-based Video Editing. In *Proc. ICCV*, 2023.

[7] M. Geyer et al. TokenFlow: Consistent Diffusion Features for Consistent Video Editing. In *Proc. ICLR*, 2024.

[8] L. Khachatryan et al. Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators. In *Proc. ICCV*, 2023.

[9] Y. Guo et al. AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning. In *Proc. ICLR*, 2024.

[10] L. Zhang, A. Rao, and M. Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *Proc. ICCV*, 2023.

[11] C. Mou et al. T2I-Adapter: Learning Adapters to Dig out More Controllable Ability for Text-to-Image Diffusion Models. In *Proc. AAAI*, 2024.

[12] H. Ye et al. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. *arXiv:2308.06721*, 2023.

[13] S. Yang et al. Rerender A Video: Zero-Shot Text-Guided Video-to-Video Translation. In *Proc. ACM SIGGRAPH Asia*, 2023.

[14] X. Li et al. VidToMe: Video Token Merging for Zero-Shot Video Editing. In *Proc. CVPR*, 2024.

[15] H. Ye and D. Xu. TaskPrompter: Spatial-Channel Multi-Task Prompting for Dense Scene Understanding. In *Proc. ICLR*, 2023.

[16] B. Zoph et al. ST-MoE: Designing Stable and Transferable Sparse Expert Models. *arXiv:2202.08906*, 2022.

[17] N. Du et al. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts. In *Proc. ICML*, 2022.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[19] R. Zhang et al. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proc. CVPR*, 2018.

[20] A. Radford et al. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. ICML*, 2021.

[21] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proc. NeurIPS*, 2017.

[22] K. C. K. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy. BasicVSR: The Search for Essential Components in Video Super-Resolution and Beyond. In *Proc. CVPR*, 2021.

[23] C. Rota, M. Buzzelli, and J. van de Weijer. Enhancing Perceptual Quality in Video Super-Resolution through Temporally-Consistent Detail Synthesis using Diffusion Models. In *Proc. ECCV*, 2024.

[24] S. Zhou, C. Li, K. C. K. Chan, and C. C. Loy. ProPainter: Improving Propagation and Transformer for Video Inpainting. In *Proc. ICCV*, 2023.

[25] Z. Li, C.-Z. Lu, J. Qin, C.-L. Guo, and M.-M. Cheng. Towards an End-to-End Framework for Flow-Guided Video Inpainting. In *Proc. CVPR*, 2022.

[26] X. Li, H. Xue, P. Ren, and L. Bo. DiffuEraser: A Diffusion Model for Video Inpainting. *arXiv:2501.10018*, 2025.