Sign Language Video Synthesis via Loss-Guided Multi-Expert GANs (Technical Report)

Nong Dingzhan[1], REN Zhihao[1], Li Ziqi[1], and Tim Lo[*1]

[1]Glassbox AI

nongdingzhan@gmail.com, renzhihao0919@gmail.com, ziq93812@gmail.com, tim.lo@xai.hk

Abstract

This preliminary technical report presents a framework for sign language video synthesis using a loss-guided multi-expert Generative Adversarial Network (GAN) to enhance communication for individuals with hearing impairments. Three specialized discriminators—global, hand, and head—each guide a corresponding expert branch in the generator toward a distinct visual region, enabling implicit feature specialization without explicit diversity losses. To stabilize this multi-discriminator system, whose early-phase training otherwise exhibits chaotic dynamics, we introduce a United Loss consensus mechanism that regularizes each discriminator toward the ensemble average at a 10% weight. Each branch further adopts a dual-pathway convolutional-transformer design with learnable AdaptiveFeatureFusion, balancing the stability of convolutions against the detail of windowed self-attention. The generator is trained using an alternating three-mode schedule (discriminator, holistic generation, branch-specialized generation). On a custom 156GB dataset with a filtered test set that removes easy and repetitive samples, our 0.2B-parameter variant achieves 29.8 PSNR and the 1.3B-parameter variant achieves 30.7 PSNR, with inference VRAM footprints of 1.5 GB and 8 GB respectively, enabling deployment on consumer-grade hardware. Full ablation studies remain ongoing due to the 2–3 month training cycle on a single GPU. The system was showcased at the 2025 Hong Kong Frontier Technology Summit.

1 Introduction

Sign language video synthesis is critical for improving communication accessibility for individuals with hearing impairments, requiring high-fidelity video generation with precise hand gestures and facial expressions. While Generative Adversarial Networks (GANs) have advanced image and video synthesis, traditional single-discriminator architectures struggle to capture fine-grained details in complex scenes like sign language videos. This limitation often results in unrealistic outputs, particularly for intricate hand and face movements.

To address these challenges, we propose a Multi-Discriminator GAN (MD-GAN) framework with specialized discriminators for global coherence, hand gestures, and facial expressions. This approach ensures detailed supervision of region-specific features. Additionally, we introduce a Multi parallel U-Net generator, where three parallel encoder-decoder branches focus on distinct visual regions (full body, hands, and face), with each branch driven by its corresponding discriminator to encourage implicit feature specialization.

The use of multiple discriminators can introduce training instability due to conflicting optimization objectives. To mitigate this, we propose a United Loss function that establishes a soft consensus constraint across discriminators. Each discriminator's loss is blended with the ensemble average at a 10% weight, penalizing extreme deviation from the group while preserving individual specialization. This mechanism promotes stable convergence without sacrificing the benefit of region-specific feedback.

To ensure realistic outputs, we incorporate Adaptive Instance Normalization (AdaIN) to fuse skeleton structure with style image appearance at each encoder layer. Our model is designed for consumer-grade hardware: all variants train on a single GPU and infer within 1.5–8 GB VRAM.

Our contributions include: (1) a loss-guided multi-expert framework in which specialized discriminators for hand, face, and global regions each drive a corresponding generator branch; (2) a Multi parallel U-Net framework with a dual-pathway architecture—each branch independently processes features through parallel convolutional and transformer pathways fused via learnable

*Corresponding author: Tim Lo, admin@xai.hk

weights (AdaptiveFeatureFusion), enabling stable, region-specific feature extraction; (3) a United Loss function for federated discriminator consensus; and (4) an alternating three-mode training schedule that decouples holistic generation from branch-specialized optimization. We report PSNR results across three model scales on a filtered test set that explicitly targets challenging cases.

The paper is organized as follows: Section 2 reviews related work, Section 3 details the proposed methodology, Section 4 presents experimental results, and Section 5 concludes with limitations and future directions.

2 Related Work

2.1 Generative Adversarial Networks for Video Synthesis

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have revolutionized the field of generative modeling. GANs consist of two neural networks—the Generator (G) and the Discriminator (D)—trained adversarially, where G generates synthetic data while D distinguishes between real and fake samples. This framework has inspired numerous advancements in deep learning, leading to improved architectures, training stability, and applications across various domains. A series of updates related to GAN have been introduced, like DCGAN (Radford et al., 2016), WGAN (Arjovsky et al., 2017), Cycle GAN (Zhu et al., 2017), and especially, the Style GAN series (Karras et al., 2019) (Karras et al., 2020) (Karras et al., 2021).

2.2 Style GAN

The Style GAN series advances image synthesis in GANs. Style GAN (Karras et al., 2019) introduces a generator inspired by style transfer, separating high-level attributes from stochastic variations with scale-specific control. Style GAN2 (Karras et al., 2020) improves image quality by fixing artifacts through redesigned normalization and regularization. Style GAN3 (Karras et al., 2021) ensures translation and rotation equivariance by addressing aliasing, suiting video applications.

2.3 Multi Parallel U-Net

In the realm of advanced image processing, Zhang et al. (Al Jowair et al., 2023) propose the Multi Parallel U-Net, a novel architecture that enhances the Mixture-of-Experts (MoE) paradigm for computer vision tasks. Unlike conventional MoE frameworks that rely on dynamic expert selection, the Multi Parallel U-Net activates all expert branches simultaneously for every input, ensuring uniform feature extraction and balanced gradient flow. While our architecture draws inspiration from this fully-parallel paradigm, it differs in two key respects: (1) our branches share identical input resolution rather than employing pre-trained heterogeneous encoders, and (2) our design introduces a United Loss consensus mechanism to coordinate training across branches, which is not present in the original Multi Parallel U-Net. Our v4 architecture further departs from the standard Multi Parallel U-Net by adopting a dual-pathway design within each branch, where parallel convolutional and Swin Transformer streams (Liu et al., 2021) are fused via learnable weights rather than fixed aggregation, enabling dynamic, per-branch specialization.

2.4 Multi Discriminators

In traditional GAN training, the generator tends to overly focus on partial modes of the real data distribution (e.g., repeatedly generating similar samples) while neglecting diversity, resulting in outputs that lack the high complexity of real data. However, existing classic solutions face limitations: methods like GMAN and D2GAN train discriminators independently without collaborative specialization, often yielding redundant feedback.

By contrast, in MCL-GAN (Choi and Han, 2022), each discriminator automatically concentrates on specific subsets of the dataset (e.g., distinct facial poses or texture regions in CelebA) through the MCL loss function, establishing an "expert specialization" pattern. This approach demonstrates robust generalization capabilities even in low-sample scenarios.

Our task involves sign language generation, where the data distribution complexity for facial and hand details significantly exceeds that of other components. While a multi-discriminator architecture aligns with these inherent requirements, our dataset features locally labeled guidance. Thus, we pioneer a novel training methodology for multi-discriminators within this framework.

3 Methodology

3.1 Multi-Discriminator

The Multi-Discriminator GAN (MD-GAN) comprises three discriminators: a Global Discriminator for overall video coherence, a Hand Discriminator

for gesture details, and a Head Discriminator for facial expressions, ensuring precise supervision of sign language video synthesis.

Global Discriminator The Global Discriminator processes 448×448 images across five resolution levels (448, 224, 112, 56, 28), using Haar wavelet transforms to decompose inputs into frequency subbands (6 to 24 channels). It employs MiniBatch Standard Deviation (MBSD) to prevent mode collapse and outputs a 30×30 feature map via a PatchGAN structure to enhance detail supervision.

Local Discriminators Local discriminators operate on 112×112 patches across three scales (112, 56, 28). The Hand Discriminator extracts hand regions using keypoint-based localization, scaling to 112×112 with center-crop alignment, focusing on finger positions and hand shapes. The Head Discriminator processes facial regions using nose keypoints with a 96-pixel radius, capturing expression details.

Haar Wavelet Transforms The discriminator employs Haar wavelet transforms to decompose RGB images into frequency subbands, expanding the input channels from 6 to 24 (6×4 subbands) to simultaneously capture structural (low-frequency) and textural (high-frequency) features. Additionally, it integrates MiniBatch Standard Deviation (MBSD) to compute statistical variations across mini-batches, enhancing discriminative power and preventing mode collapse by providing batch-level contextual information for improved real/fake sample distinction.

3.2 Generator

Multi-Branch Parallel Processing Architecture Multi parallel U-Net builds upon the Mixture-of-Experts (MoE) paradigm by introducing a fully parallel architecture that addresses expert load imbalance and training instability inherent in conventional MoE models. Unlike dynamic expert selection in MoE, Multi parallel U-Net simultaneously activates all branches for every input, ensuring uniform feature extraction coverage and balanced gradient flow. Moreover, each branch is driven by its corresponding specialized discriminator (hand, face, or global), which provides implicit pressure for the branch to specialize on specific visual regions without requiring explicit diversity losses. Adaptive Instance Normalization (AdaIN) is introduced to implement the image translation task in the form of style transfer. The overall encoder structure is illustrated in Figure 1.

All three branches share the same 448×448 input. To ensure that each branch develops distinct feature representations despite identical inputs, we adopt a **two-stage dual-pathway architecture** within each branch, illustrated in Figure 3.

Stage 1 — channel mixing vs. spatial convolution: The branch input x is processed through three parallel streams: (i) a residual path (1×1 convolution followed by GroupNorm), (ii) a ConvTransformerBlock—a lightweight transformer that operates along the channel dimension via channel-wise MLP with LayerNorm, capturing cross-channel dependencies without the quadratic cost of spatial self-attention, and (iii) a 1×1 linear projection followed by GroupNorm and an Attn_Conv2d block (convolution with optional Local-Global Merged Attention). The ConvTransformerBlock stream and the linear-projection/Attn_Conv2d stream are then fused via an AdaptiveFeatureFusion (AFF) module.

Stage 2 — local convolution vs. windowed self-attention: The fused features again split into two parallel streams. One stream continues through a purely convolutional Attn_Conv2d block (attention disabled). The other stream is patch-embedded via PatchEmbed, processed by a Swin Transformer block with Window-based Multi-head Self-Attention (W-MSA), and unpatched back to the spatial domain. A second AFF module fuses these two streams. A final GroupNorm and the additive residual connection complete the block, followed by 2× average-pooling downsampling.

Why Dual-Pathway Fusion Works: Each encoder block (Downsample_Vit) implements the two-stage dual-pathway design described above and illustrated in Figure 3. The AFF fusion mechanism computes a soft weighted combination of its two input streams:

$$\text{Fused} = \alpha \cdot \text{Stream}_1 + (1 - \alpha) \cdot \text{Stream}_2,$$

where α is produced by applying softmax or sigmoid to a learnable parameter, allowing the model to dynamically balance the contribution of each stream. This is not a fixed residual addition—the fusion weights are optimized during training and can vary across branches and depth levels. This two-stage design exploits the complementary strengths of the two streams. In our experiments, we observed a consistent trade-off: the convolu-

Figure 1: Encoder architecture with three parallel branches (head, hand, global). Each branch processes identical input through independent dual-pathway Downsample_Vit blocks. AdaIN fuses content and style streams at each level to produce skip connections.

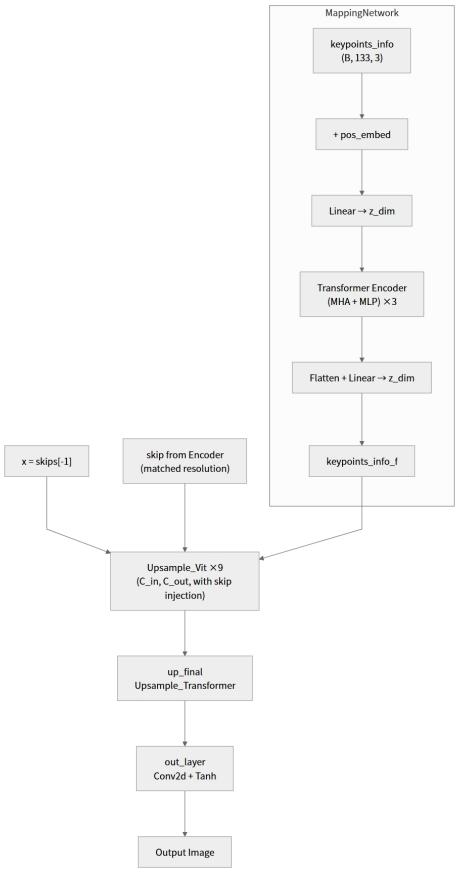

Figure 2: Decoder architecture. Three parallel encoder-decoder branches produce skip connections, concatenated and processed by Upsample_Vit blocks with keypoint-guided cross-attention.

tional stream produces globally stable but visually smooth features, while the Swin Transformer stream captures finer detail but exhibits instability at high-frequency boundaries—such as clothing edges, finger contours, and face-background transitions. The learnable AFF fusion resolves this tension: the convolutional stream acts as a stabilizing anchor that suppresses boundary jitter, while the Transformer stream injects the fine detail that pure convolution lacks. Because the fusion weights are optimized independently per branch and depth level, the stability-detail balance adapts dynamically, and the divergent discriminator-driven gradients prevent homogenization across the three parallel branches. We attribute this complementarity to the opposing inductive biases of the two operations under adversarial training: convolution acts as a low-pass filter that accumulates smoothing across layers, yielding stable but over-smoothed features, while windowed self-attention applies sharp, content-dependent weights that preserve high-frequency detail at the cost of sensitivity to boundary perturbations. Under discriminator pressure—which penalizes both blur and boundary instability—the two streams are pushed toward opposite extremes, and the learnable AFF fusion restores the balance. This learnable two-stage fusion replaces the fixed scalar blending of Swin features used in earlier iterations, where a constant weight combined the two streams. The scalar approach proved insufficient to balance the opposing stability–detail trade-off described above, motivating the fully learnable AdaptiveFeatureFusion design.

Decoder Upsample Blocks: The decoder employs a symmetric architecture (Figure 2) with Upsample_Vit blocks (Figure 4) that mirror the encoder's dual-pathway design. Each Upsample_Vit block first upsamples the incoming feature map via bilinear interpolation, then concatenates the corresponding skip connection from the encoder—a concatenation of the AdaIN-fused head, hand, and global features. The combined features are split into three MoE branches (head, hand, global), each processed independently through a conv_linear projection followed by parallel CNN and Swin Transformer pathways fused via AdaptiveFeatureFusion. At select layers (up_3, up_5, up_7), keypoints_info_f from the MappingNetwork is injected through a SpatialTransformer and CrossSwinTransformerBlock, enabling keypoint-guided cross-attention. After MoE processing, the three branches are concatenated and passed through a second, global dual-pathway stage (PatchEmbed $\rightarrow$ Swin $\rightarrow$ AdaptiveFeatureFusion_all), followed by convolutional refinement with Local-Global Merged Attention.

The final block, Upsample_Transformer (Figure 5), is a simplified single-channel variant that omits MoE branches, skip injection, and cross-attention. It operates at the original resolution (no upsampling) and serves purely as a refinement stage: a residual path preserves the input, while the main path processes features through parallel CNN and Swin Transformer pathways fused via AdaptiveFeatureFusion, followed by two convolutional layers with intermediate attention.

MappingNetwork: Keypoints are encoded by a lightweight MappingNetwork consisting of a Transformer encoder (3 layers, 8 heads) with positional embeddings. The input—133 skeleton joints with 3D coordinates—is projected to a latent dimension z_{dim}, processed through multi-head self-attention and MLP blocks, then flattened and linearly pro-

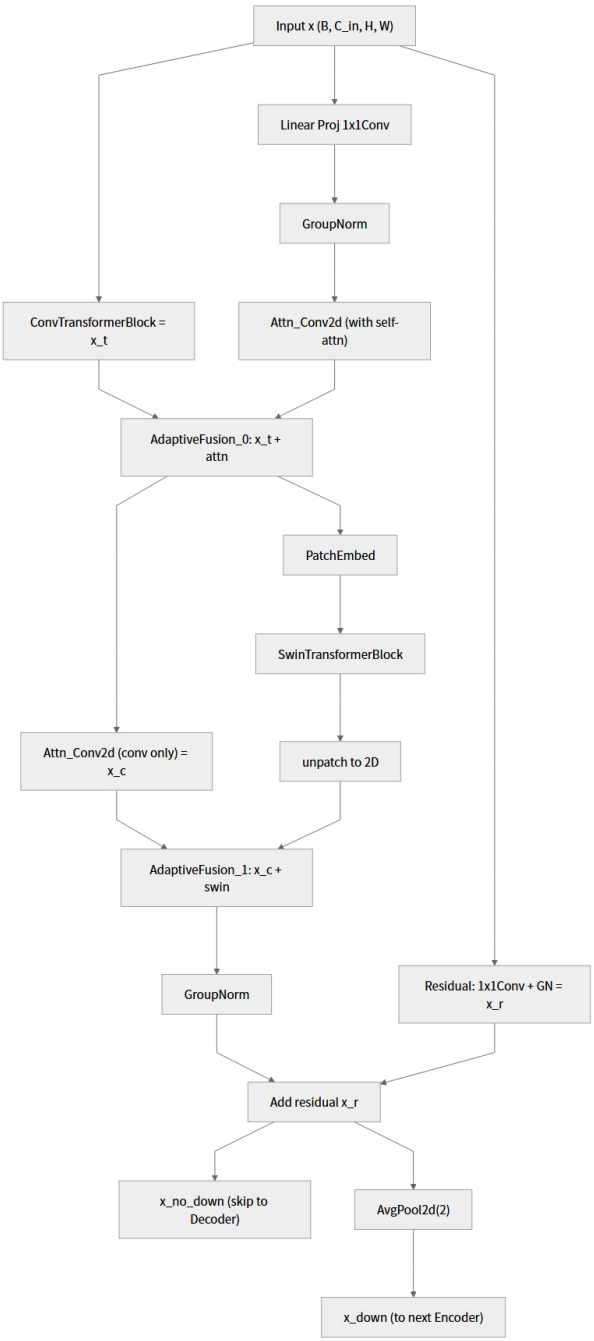

Figure 3: Downsample_Vit: dual-pathway convolutional-transformer block. Each encoder layer employs a two-stage design: Stage 1 fuses a ConvTransformer stream with a convolutional-attention stream, and Stage 2 fuses a pure-convolution stream with a Swin Transformer stream, each via learnable AdaptiveFeatureFusion, with a residual connection and $2\times$ downsampling.

jected to produce keypoints_info_f, which is shared across all decoder layers.

Local-Global Merged Attention: We introduce a multi-scale attention mechanism that integrates local and global feature representations to improve contextual awareness (Figure 6). The proposed approach comprises three components:

Local Attention: Following the ViT paradigm, high-resolution feature maps are divided into 14×14 patches. Query-Key-Value (QKV) computations are performed to capture local attention, and the results are reassembled to the original resolution.

Sub-Local Attention: High-resolution feature maps are downsampled to 112×112 using average pooling and partitioned into 28×28 patches. After QKV computations, the patches are unpatched, and nearest-neighbor interpolation is applied to upsample the features back to the original resolution.

Global Attention: The high-resolution feature maps are further downsampled to 56×56 via average pooling. QKV computations are performed, followed by nearest-neighbor interpolation to restore the original resolution.

Merged Attention: The final attention map is computed as a weighted combination:

$$\begin{aligned}\text{Merged Attention} = {} & 0.85 \times \text{Local} \\ & + 0.1 \times \text{Sub-Local} \\ & + 0.05 \times \text{Global}.\end{aligned}$$

This formulation balances fine-grained local details with broader contextual information. The 0.85/0.10/0.05 weighting reflects our design principle that local detail should dominate, with sub-local and global context serving as soft constraints. This attention is instantiated as a dedicated block—Conv2D with Local-Global Merged Attention—used within Attn_Conv2d modules (Figure 7).

AdaIN-based Style-Skeleton Fusion: The model takes as input two three-channel images: a skeleton image and a style image. Both are processed through separate encoders. At each encoder layer, the outputs are combined using Adaptive Instance Normalization (AdaIN). The resulting features serve as skip connections in a U-Net architecture, facilitating the integration of structural (skeleton) and stylistic (appearance) information across multiple scales. This design ensures that the generated video maintains both accurate pose structure and photorealistic texture.

3.3 United Loss and Training Strategy

In this work, we introduce a GAN framework that incorporates a generator and multiple discriminators. Training multiple discriminators independently poses a severe stability challenge that is most acute in the early stages of training. During this phase, gradients are large, and the three discriminators—each optimizing toward its own realism criterion—produce conflicting gradient directions. Without coordination, the generator receives incoherent feedback and frequently fails to converge, with output quality degrading progressively over successive iterations rather than improving.

Notably, we observe that this conflict is transient: as training progresses and gradients shrink, the discriminators gradually develop an implicit coordination—an emergent "consensus" where their individual criteria naturally align. At this later stage, the need for an explicit consensus mechanism weakens, since the discriminators have effectively learned to cooperate without external constraint. United Loss therefore functions primarily as a *training wheel* that stabilizes the vulnerable early phase, while its influence naturally recedes once coordination emerges.

To provide this early-phase stability, we propose a United Loss function that unifies the training objectives of all discriminators through a soft consensus mechanism. Each discriminator computes its own adversarial loss independently, but is additionally regularized by the ensemble average of all three discriminators at a weighting of 10% (i.e., $\lambda_{\text{united}} = 0.1$). Concretely, for each discriminator D_i:

$$\mathcal{L}_{D_i}^{\text{total}} = \mathcal{L}_{D_i}^{\text{adv}} + \lambda_{\text{united}} \cdot \mathcal{L}_{\text{united}},$$

where $\mathcal{L}_{\text{united}}$ is computed from the equally-weighted average output of all three discriminators. This formulation ensures that no single discriminator can deviate too far from the ensemble consensus during the unstable early phase, while still preserving individual specialization. The 10% weight was chosen empirically to balance stability (higher weight) against discriminative diversity (lower weight).

The efficacy of this simple averaging mechanism can be understood through the dynamics of discriminator divergence. Without United Loss, we observe that the generator tends to *capitulate* to a single discriminator—optimizing toward whichever criterion is momentarily easiest to satisfy. This creates a runaway feedback loop: the favored discrim-

Figure 4: Upsample_Vit: dual-pathway decoder block with skip injection.

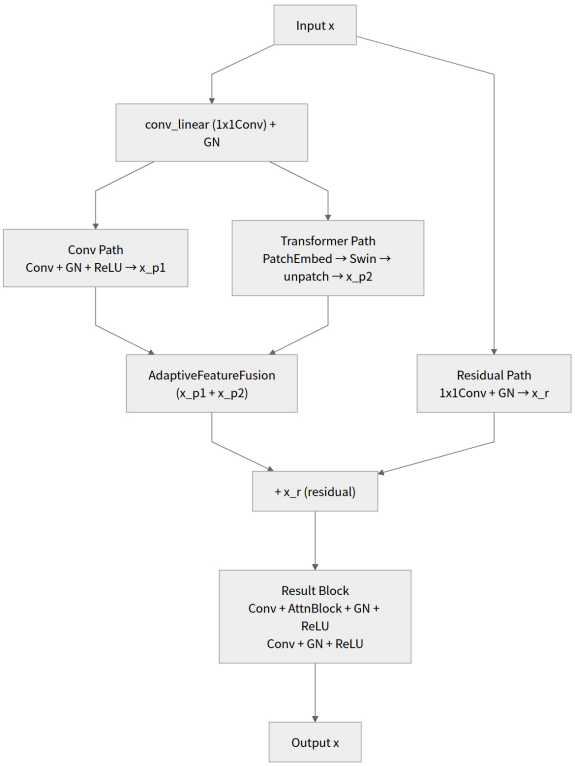

Figure 5: Upsample_Transformer: final decoder block.

inator's loss collapses to a very low value, while the remaining discriminators' losses surge, since the generator no longer attends to their criteria. The result is de facto single-discriminator training, with the other two discriminators contributing only noise.

United Loss breaks this loop through a *proportional penalty* on deviation from consensus. Since $\mathcal{L}_{\text{united}}$ is the equally-weighted average of all three losses, a discriminator whose individual loss has collapsed contributes a disproportionately large share of its own total loss through the consensus term: with $\mathcal{L}_{D_i}^{\text{adv}}$ near zero, the total loss $\mathcal{L}_{D_i}^{\text{total}} \approx \lambda_{\text{united}} \cdot \mathcal{L}_{\text{united}}$ is dominated entirely by the ensemble average, which is elevated by the other (high-loss) discriminators. Conversely, a discriminator with an already-high loss is relatively unaffected, because its total loss is dominated by its own adversarial term rather than the consensus term. The low-loss "culprit" is therefore penalized the most, in proportion to its deviation from the group, while the high-loss "victims" remain free to exert their own gradients. This asymmetry pulls the deviating discriminator back toward consensus without suppressing the legitimate specialization signals of the others.

To integrate feedback from all three discriminators, we define a weighted sum of their individual losses for the generator:

$$\mathcal{L}_{\text{united}}^{\text{gen}} = \lambda_g \, \mathcal{L}_{\text{global}} + \lambda_h \, \mathcal{L}_{\text{hand}} + \lambda_f \, \mathcal{L}_{\text{head}},$$

where $\lambda_g = 0.33$, $\lambda_h = 0.33$, and $\lambda_f = 0.33$. Each loss term is computed via binary cross-entropy (BCE) on real versus generated samples.

Alternating Training Modes The training loop cycles through three fixed modes in a deterministic rotation. Let s denote the training step; the active mode is determined by

$$\text{mode}(s) = \lfloor s/10 \rfloor \bmod 3,$$

yielding a 1:1:1 rotation in which each mode persists for 10 consecutive steps:

1. **Discriminator mode** (mode $= 0$): All three discriminators are updated in parallel while the generator is frozen. If a sample contains no hands, only the global discriminator is updated; the hand and head discriminators receive no gradient on such samples.

2. **Generator overall mode** (mode $= 1$): The aggregated generator loss $\mathcal{L}_G$ is backpropagated through all generator parameters simultaneously, providing holistic guidance from the combined global and local feedback.

3. **Generator partial mode** (mode $= 2$): Each branch (head, hand, global) is updated independently using its own discriminator-specific loss and its own optimizer, enabling region-targeted optimization without cross-branch gradient conflict. For samples without hands, this mode degenerates to a global update, since the hand branch has no valid input to process.

This deterministic rotation replaces an earlier cosine-scheduled phased-freezing design. We found empirically that cosine scheduling—whose gradual re-weighting of phases introduced an additional degree of freedom—yielded less stable convergence than the fixed 1:1:1 alternation, which imposes a rigid, predictable rhythm on the adversarial dynamics.

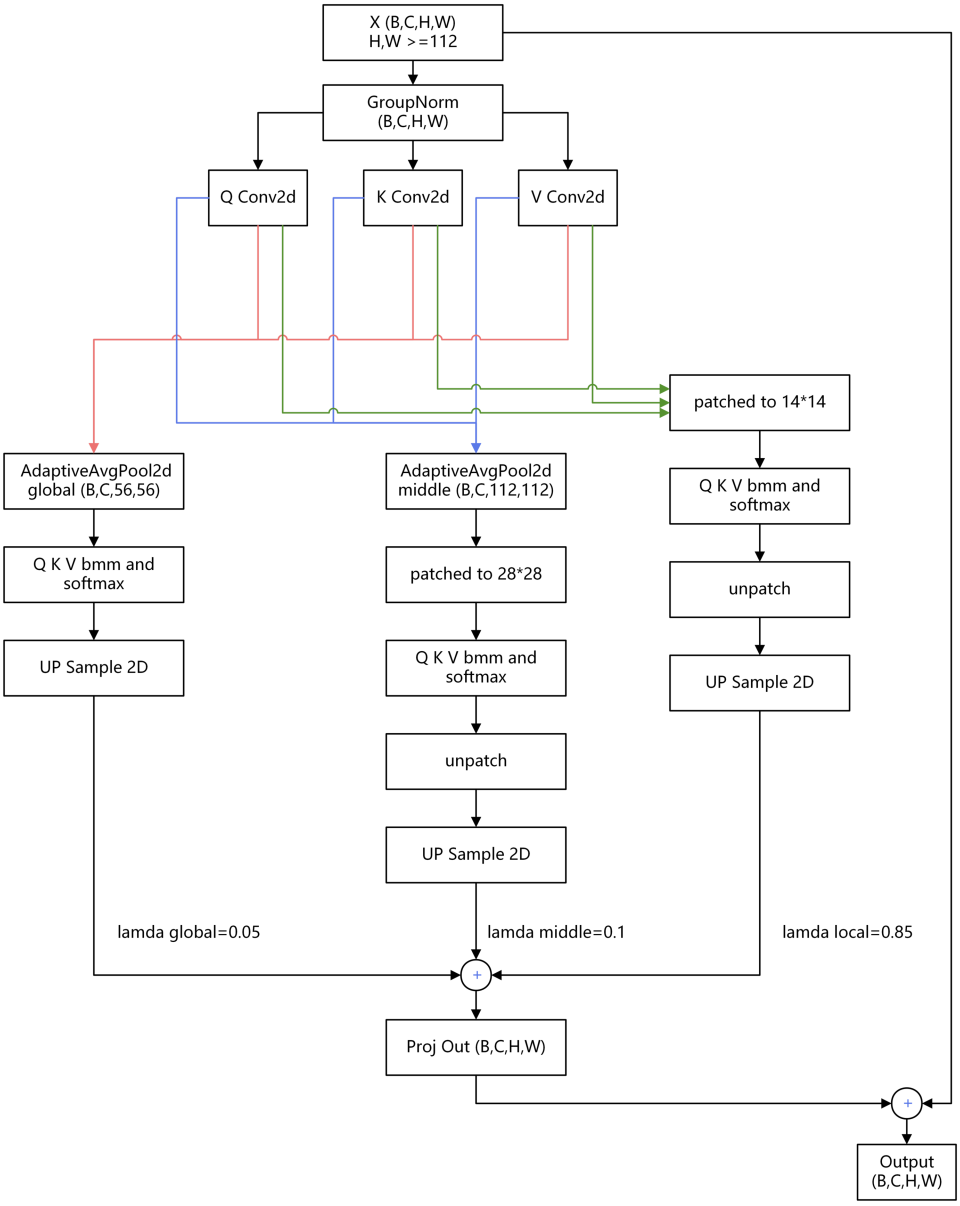

Figure 6: Local-Global Merged Attention block.

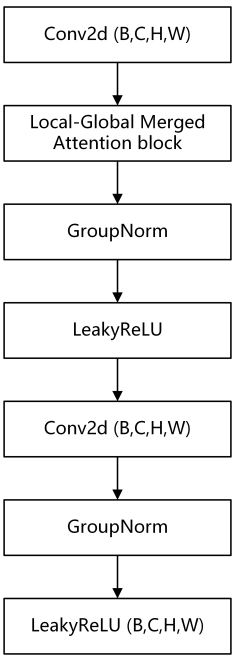

Figure 7: Conv2D with Local-Global Merged Attention block (B,C,H,W).

Discriminator United Loss For a real sample x and a generated sample $\hat{x}$, let

$$\big\{D(x),\, D(\hat{x}),\, D_{\text{hand}}(x),\, D_{\text{hand}}(\hat{x}),\, D_{\text{head}}(x),\, D_{\text{head}}(\hat{x})\big\}$$

be the outputs of the three discriminators. After sampling real labels $r_{\text{real}} \sim \mathcal{U}(0.99, 1.0)$ and fake labels $r_{\text{fake}} \sim \mathcal{U}(0.0, 0.01)$, the combined discriminator loss incorporates the United Loss term:

$$y^{\text{U}}_{\text{real}} = \frac{1}{3}\,D(x) + \frac{1}{3}\,D_{\text{hand}}(x) + \frac{1}{3}\,D_{\text{head}}(x) \quad (1)$$

$$y^{\text{U}}_{\text{fake}} = \frac{1}{3}\,D(\hat{x}) + \frac{1}{3}\,D_{\text{hand}}(\hat{x}) + \frac{1}{3}\,D_{\text{head}}(\hat{x}) \quad (2)$$

$$\mathcal{L}^{\text{U}}_{\text{disc}} = \mathbb{E}\big[\ell_{\text{BCE}}(y^{\text{U}}_{\text{real}},\, r_{\text{real}})\big] + \mathbb{E}\big[\ell_{\text{BCE}}(y^{\text{U}}_{\text{fake}},\, r_{\text{fake}})\big] \quad (3)$$

Generator United Loss Let $\hat{x}$ denote a generated sample. Define the fused discriminator score for the generator as

$$y_{\text{gen}} = \frac{1}{3}\,D(\hat{x}) + \frac{1}{3}\,D_{\text{hand}}(\hat{x}) + \frac{1}{3}\,D_{\text{head}}(\hat{x}). \quad (4)$$

Using a randomly sampled "real" label $r \sim \mathcal{U}(0.99, 1.0)$, the United Loss for the generator is

$$\mathcal{L}^{\text{U}}_{\text{gen}} = \mathbb{E}\big[\ell_{\text{BCE}}(y_{\text{gen}}, r)\big], \quad (5)$$

where ℓ_{BCE} denotes the binary cross-entropy loss. This equal-weight fusion prevents any single discriminator from dominating gradient direction, improving generation quality and fairness.

4 Experiments

4.1 Dataset

We train and evaluate on a custom 156 GB dataset of sign language videos. Each video depicts a single vocabulary word, recorded with a consistent three-part structure: the signer begins in a resting pose (hands clasped in front of the abdomen), transitions into the sign language gesture, and returns to the resting pose. The resting-pose frames—which we denote as *easy samples*—account for approximately 50% of each video's duration on average, and hence roughly half of the dataset by frame count. These easy samples are highly repetitive and trivially learnable: a model that simply reproduces the resting pose achieves near-perfect reconstruction on them, inflating PSNR without reflecting the ability to generate meaningful gestures.

We therefore construct a **filtered test set** that excludes all easy samples, ensuring that every reported metric targets the challenging transition and gesture frames where generation is prone to collapse or artifacts. For training, we likewise retain only a small fraction of easy samples—sufficient for the model to learn the resting pose itself, but not enough to dominate the optimization objective. Each sample consists of a skeleton image (extracted via pose estimation), a style image (for appearance transfer), and the corresponding ground-truth video frame. When evaluated on the full unfiltered test set including easy samples, PSNR exceeds 36, but we consider the filtered metric far more informative for assessing generation quality on difficult poses and hand configurations.

4.2 Training Configuration

All models are trained on a single consumer GPU (NVIDIA RTX 4090, 24 GB or 48 GB variant) using the AdaBelief optimizer. Training employs a cosine learning rate schedule with linear warmup over the first 50,000 steps. Full convergence requires approximately 5–6 million steps (2–3 months of training), with PSNR exhibiting a characteristic non-linear progression: extended plateaus punctuated by phase-transition breakthroughs, followed by terminal oscillation. We hypothesize that the plateau-to-breakthrough dynamics reflect the multi-objective optimization landscape induced by three competing discriminators with United Loss consensus—when one branch discovers a beneficial feature representation, the consensus mechanism propagates improvements to the ensemble,

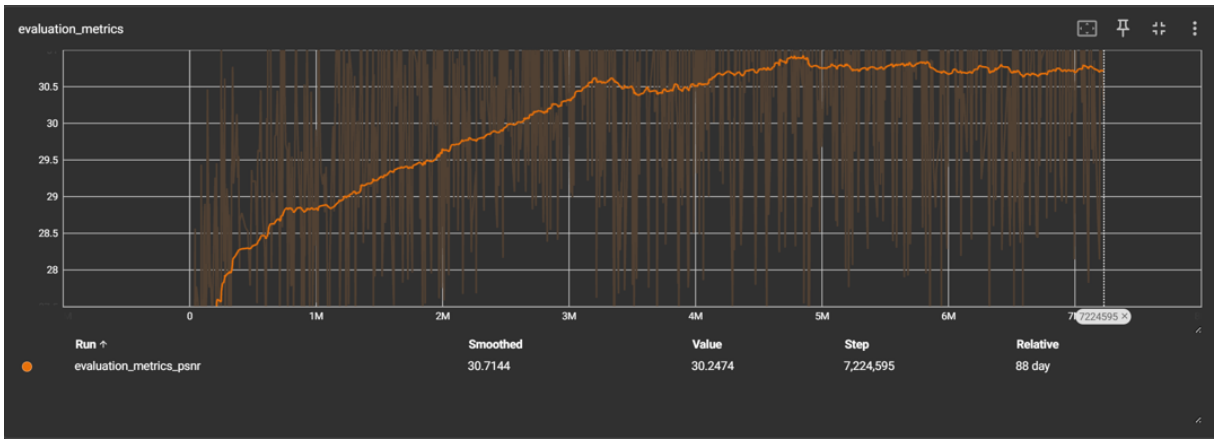

Figure 8: PSNR progression of the 1.3B model over approximately 7M training steps on a single RTX 4090 (convergence is achieved by 5–6M steps; training continues to 7M to verify stability). The curve exhibits three characteristic phases: (1) rapid initial improvement during the first ∼500k steps, (2) an extended plateau punctuated by abrupt breakthroughs—a signature of United Loss consensus dynamics where improvements in one branch propagate through the ensemble, and (3) terminal oscillation after convergence, consistent with GAN training dynamics.

triggering cascading breakthroughs (Figure 8).

4.3 Results: Model Scale vs. PSNR

We evaluate three model scales, all sharing the same architecture but varying in channel width:

Model	Params	PSNR	VRAM
Small	0.2B	29.8	1.5 GB
Medium	0.66B	30.4*	∼5 GB
Large	1.3B	30.7	8 GB

Table 1: PSNR results across model scales. All values evaluated on the filtered test set (challenging cases only). *The 0.66B result is from an earlier dataset version (v3, lower contrast); re-training on v4 is in progress and expected to yield 30.1–30.2. The 0.2B and 1.3B results are from the v4 dataset.

The monotonic PSNR improvement from 29.8 to 30.7 as parameters increase 6.5× (0.2B to 1.3B) suggests that the architecture effectively utilizes additional capacity rather than overfitting or suffering from optimization degradation. All three variants remain deployable on consumer-grade hardware, with inference VRAM ranging from 1.5 GB to 8 GB.

4.4 Training Dynamics: Qualitative Observations

While a controlled ablation of United Loss on the current v4 architecture has not been completed (requiring a separate 2–3 month training run), we report observational evidence consistent with the hypothesized consensus mechanism:

1. **Non-linear convergence**: Training exhibits plateau-to-breakthrough phase transitions rather than smooth PSNR improvement. We observe that breakthroughs in one region (e.g., hand clarity) tend to coincide with improvements in other regions within 50,000–100,000 subsequent steps, consistent with United Loss propagating quality signals across the discriminator ensemble.

2. **Branch specialization without collapse**: Despite all three branches receiving identical 448×448 inputs, qualitative inspection of intermediate outputs confirms that the hand branch generates sharper hand regions, the face branch produces more expressive facial details, and the global branch maintains better body coherence. This implicit specialization is driven purely by discriminator-specific loss signals—the dual-pathway architecture with learnable fusion and independent optimizers prevent gradient homogenization.

3. **Stability at scale**: Training remains stable across all three model scales without requiring gradient penalty, spectral normalization, or other common stabilization techniques beyond United Loss. We attribute this to the consensus mechanism's regularizing effect on individual discriminator updates.

4.5 Planned Ablation: United Loss Contribution

A controlled ablation comparing training with and without United Loss on the v4 architecture is planned but has not yet been executed due to the prohibitive training time (2–3 months per run on a single consumer GPU). The experimental design is as follows:

- **Baseline (with United Loss, $\lambda_{\text{united}} = 0.1$)**: Already completed; PSNR 29.8 (0.2B) / 30.7 (1.3B).
- **Ablation (without United Loss, $\lambda_{\text{united}} = 0$)**: Planned. We hypothesize that removing United Loss would lead to (1) increased training oscillation due to unconstrained discriminator divergence, (2) potential branch homogenization as discriminators provide conflicting gradient signals, and (3) reduced final PSNR, consistent with observations from an earlier ablation study conducted during the early development of the v4 architecture, where United Loss demonstrated a measurable stabilizing effect.
- **Varied λ_{united}**: Planned. Exploring the trade-off between consensus strength (higher λ) and discriminative diversity (lower λ).

5 Conclusion

5.1 Summary of Contributions

We have presented a loss-guided multi-expert GAN framework for sign language video synthesis featuring: (1) three specialized discriminators (global, hand, face) with Haar wavelet preprocessing; (2) a Multi parallel U-Net generator where three independent encoder-decoder branches specialize on distinct visual regions, driven by corresponding discriminators; (3) a United Loss consensus mechanism that stabilizes multi-discriminator training through a 10% soft blending of individual losses with the ensemble average; (4) a dual-pathway convolutional-transformer architecture with AdaptiveFeatureFusion—a learnable blending mechanism that fuses parallel convolutional and Swin Transformer streams at each encoder block, ensuring feature divergence across branches; (5) a Local-Global Merged Attention mechanism integrating three spatial scales; and (6) an alternating three-mode training schedule (discriminator / holistic / branch-specialized) with independent per-branch optimizers. On a filtered test set targeting challenging cases, our architecture achieves 29.8–30.7 PSNR across model scales, with inference deployable on consumer-grade hardware.

5.2 Limitations

This work has several limitations that we aim to address in future iterations:

Training Cost: Full training requires 2–3 months on a single consumer GPU, which has precluded comprehensive ablation studies of individual architectural components (United Loss, dual-pathway architecture, alternating training modes, multi-scale attention). Each ablation would require a full retraining cycle, making systematic component-wise analysis infeasible under current resource constraints.

Ablation Gap: The contribution of United Loss to training stability and final PSNR has not been isolated on the current v4 architecture. While observational evidence—non-linear convergence dynamics, maintained branch specialization, and training stability without external regularization—is consistent with the hypothesized consensus mechanism, rigorous causal attribution requires controlled experiments.

Scaling Bottleneck: The current architecture activates all three branches at both training and inference time. Extending to more specialized branches (e.g., separate left-hand and right-hand experts, facial sub-region experts, or multiple domain-specific experts) would incur linear cost increases in both compute and VRAM. This scalability limitation directly motivates our ongoing work on label-routed expert selection, where a pre-trained router activates only the relevant expert branches before generation begins, maintaining bounded inference cost regardless of the number of experts.

Metric Limitations: We currently report only PSNR. Comprehensive evaluation with perceptual metrics (FID, LPIPS), action accuracy, and human evaluation is planned but not yet completed. The relationship between PSNR and perceptual quality in sign language generation—where pixel-level accuracy on hands and faces matters disproportionately—warrants further investigation.

5.3 Version History

The current architecture (v4) is the result of iterative development over approximately two years. We document the progression below. Note that

model and dataset versions evolved independently; the dataset also progressed through v1–v4, with v4 (current) increasing image contrast for improved visual appeal at a cost of approximately 0.1–0.2 PSNR relative to v3. PSNR was rigorously recorded only on v4; earlier versions were evaluated subjectively due to resource constraints.

- **v1 — AdaIN as Style Transfer**: The initial approach concatenated the skeleton (content) and style images along the channel dimension before feeding them to a standard U-Net. This caused information entanglement in deep encoder layers, where skeleton structure and style appearance became indistinguishable; the decoder could not reliably reconstruct fine details, frequently producing hands with incorrect numbers of fingers. Using two separate encoders or concatenating outputs at every layer would preserve information but double the decoder's channel count, making training infeasible under consumer GPU memory constraints. The key observation was that sign language generation could be reformulated as a *style transfer* problem: the skeleton defines spatial structure, while the style image provides appearance. Replacing concatenation with Adaptive Instance Normalization (AdaIN) at each encoder layer—processing content and style through two encoder streams that remain independent until AdaIN fusion—preserved both structural and stylistic information throughout the downsampling hierarchy without increasing decoder complexity. This dual-stream design stabilized finger generation and formed the foundation for all subsequent versions.

- **v2 — Keypoint-Guided Generalization**: Although v1 achieved stable hand generation, a generalization problem emerged. Skeleton sequences between vocabulary items were algorithmically interpolated, and coordinate normalization was necessary to prevent the person from drifting across the frame during transitions. However, this normalization introduced a train-test mismatch: the same gesture could appear at different normalized positions during training versus inference, occasionally causing the model to break down. Applying dropout to simulate coordinate noise—a standard regularization technique—produced negligible improvement, suggesting the deficiency was not in image-level robustness but in the model's *awareness of skeleton coordinates*. Inspired by StyleGAN3's use of latent codes for fine-grained control, we embedded the 133-joint 3D keypoint coordinates into the decoder via a lightweight MappingNetwork (a 3-layer Transformer encoder) and injected the resulting keypoints_info_f at select decoder layers. This gave the model explicit coordinate knowledge as a form of "informative noise," dramatically improving generalization: as long as coordinate shifts were not extreme, generation remained stable.

- **v3 — Local Hand Discriminator**: With generalization resolved, the system reached a commercially viable quality level. Deaf users found the output intelligible and satisfactory. However, fine details remained unconvincing: fingers appeared unnaturally smooth, lacking shadows and texture. We hypothesized that the global discriminator's 448×448 receptive field could not provide sufficiently granular gradient signals for small hand regions. Exploiting the skeleton coordinates already available, we cropped a 112×112 region centered on each hand and trained an additional, lightweight hand discriminator operating on these patches. The generator loss was extended to $\mathcal{L}_{\text{main}} + 0.25 \cdot \mathcal{L}_{\text{hand}}$, with the two discriminators trained fully independently. Notably, while the two discriminators exhibited mild oscillation during training, this drift remained controllable within an overall convergent trajectory. Because convergence was ultimately achieved, the potential severity of multi-discriminator dynamics was not yet apparent at this early exploratory stage—the pronounced instability that would later motivate United Loss only emerged when a third discriminator was introduced in v4.

- **v4 (current) — Multi-Expert Parallel Architecture**: With the company's immediate commercial risks resolved, we turned to a more fundamental question: on a consumer GPU (RTX 4090), was the model allocating a substantial fraction of its capacity to learning unimportant details such as clothing wrinkles and folds? We hypothesized that partitioning the generator into specialized expert

branches—each guided by a dedicated discriminator toward a distinct visual region—would concentrate representational capacity on perceptually critical areas (hands, face) while maintaining global coherence.

However, adding the third (head) discriminator immediately destabilized training. Whereas the two-discriminator system of v3—despite mild oscillation—remained within a controllable, convergent trajectory, the three-discriminator system exhibited dynamics reminiscent of the *three-body problem*: without an external consensus constraint, the three optimizers drifted into chaotic, non-convergent trajectories. This instability directly motivated the United Loss consensus mechanism described in Section 3.3, which acts as the external constraint that binds the three discriminators into a stable configuration. The resulting architecture—three encoder-decoder branches with identical 448×448 input, a dual-pathway convolutional-transformer design with AdaptiveFeatureFusion, United Loss consensus across three discriminators, and an alternating three-mode training schedule—is the first version for which we report systematic PSNR measurements: 29.8 (0.2B) and 30.7 (1.3B) on the filtered test set of dataset v4.

5.4 Future Work

Beyond completing the planned ablation studies, our primary research direction is extending the loss-guided expert specialization paradigm—validated here in the GAN framework—to diffusion models. Our ongoing proposal, *Loss-Guided MoE Diffusion with Label-Routed Experts*, aims to replace full branch activation at inference with a pre-trained label router that selects only relevant expert branches before generation begins. This would preserve the quality gains of multi-expert architectures while keeping inference cost bounded, directly addressing the scaling bottleneck identified in the current work.

5.5 Acknowledgments

The system was showcased at the 2025 Hong Kong Frontier Technology Summit. We thank the event organizers and attendees for valuable feedback.

References

Hamdan Al Jowair, Mansour Alsulaiman, and Ghulam Muhammad. 2023. Multi parallel u-net encoder network for effective polyp image segmentation. *Image and Vision Computing*, 137:104767.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *Preprint*, arXiv:1701.07875.

Jinyoung Choi and Bohyung Han. 2022. Mcl-gan: Generative adversarial networks with multiple specialized discriminators. *Preprint*, arXiv:2107.07260.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *Preprint*, arXiv:1406.2661.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. In *Proc. NeurIPS*.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. *Preprint*, arXiv:1812.04948.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *Preprint*, arXiv:1511.06434.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *Preprint*, arXiv:1703.10593.

Algorithm 1 Main Training Loop

Initialize:
G, D, D_{hand}, D_{head} ▷ Generator and Discriminators
$opt_G, opt_D, opt_{D_{hand}}, opt_{D_{head}}$ ▷ Optimizers
Hyperparameters: $\lambda_{hand}, \lambda_{head}, \lambda_{L1}, \lambda_{united}$
for each training step t **do**
    $(content, target, style) \leftarrow$ LoadBatch()
    $(bodies, faces, left\_hands, right\_hands) \leftarrow$ ProcessKeypoints(npy_list)
    $(hand\_input, real\_hands, head\_input, real\_head) \leftarrow$ CropRegions$(content, target)$
    $mode \leftarrow (\lfloor t/10 \rfloor) \bmod 3$ ▷ Fixed 1:1:1 rotation
    if $mode == 0$ **then**
        TrainDiscriminators(...)
    else if $mode == 1$ **then**
        TrainGenerator(..., overall)
    else
        TrainGenerator(..., partial)
    end if
    if $t \mod 20000 == 0$ **then**
        UpdateLearningRates()
    end if
    if $t \mod$ val_gap $== 0$ **then**
        RunValidation()
    end if
    if $t \mod$ save_gap $== 0$ **then**
        SaveCheckpoints()
    end if
end for

A Training Algorithms (Pseudocode)

Notation

- G: Generator, D: Main (global) discriminator
- D_{hand}: Hand discriminator, D_{head}: Head discriminator
- λ: Loss weights, BCE: Binary cross-entropy
- avg: Equally-weighted average of discriminator outputs

Key Features

- Alternating training modes (discriminator / generator overall / generator partial) with fixed 1:1:1 rotation
- Multi-scale discriminators (global, hand, head) with United Loss consensus
- Part-specific optimization with independent optimizers per generator branch
- Combined adversarial and L1 losses for generator
- Cosine learning rate scheduling with linear warmup
- Regular validation and checkpointing

Algorithm 2 TrainGenerator Procedure

procedure TRAINGENERATOR()
    $gen\_output \leftarrow G(content, style, bodies, faces, left\_hands, right\_hands)$
    $(gen\_hands, gen\_head) \leftarrow$ CropGeneratedRegions(gen_output)
    ▷ With discriminators frozen
    $D\_fake \leftarrow D(content, gen\_output)$
    $D\_hand\_fake \leftarrow D_{hand}(hand\_input, gen\_hands)$
    $D\_head\_fake \leftarrow D_{head}(head\_input, gen\_head)$
    $L_{GAN} \leftarrow BCE(D\_fake, 1)$
    $L_{GAN\_hand} \leftarrow \lambda_{hand} \cdot BCE(D\_hand\_fake, 1)$
    $L_{GAN\_head} \leftarrow \lambda_{head} \cdot BCE(D\_head\_fake, 1)$
    $L_{L1} \leftarrow \lambda_{L1} \cdot \|target - gen\_output\|_1$
    $L_{united} \leftarrow BCE(\text{avg}(D\_fake, D\_hand\_fake, D\_head\_fake), 1)$
    $L_G \leftarrow \sum$ losses
    opt_G.zero_grad()
    opt_{Global}.zero_grad()
    opt_{Hand}.zero_grad()
    opt_{Head}.zero_grad()
    $L_{Global} \leftarrow L_{GAN} + \lambda_{Gu} \cdot L_{united} + L_{L1}$
    $L_{hand} \leftarrow L_{GAN\_hand} + \lambda_{Gu} \cdot L_{united} + L_{L1}$
    $L_{head} \leftarrow L_{GAN\_head} + \lambda_{Gu} \cdot L_{united} + L_{L1}$
    ▷ If mode == 1 (overall):
    L_G.backward()
    opt_G.step()
    ▷ If mode == 2 (partial):
    L_{Global}.backward()
    opt_{Global}.step()
    L_{hand}.backward()
    opt_{Hand}.step()
    L_{head}.backward()
    opt_{Head}.step()
end procedure

Algorithm 3 TrainDiscriminators Procedure

procedure TRAINDISCRIMINATORS()
▷ With generator frozen
$gen\_output \leftarrow G(content, style, bodies, faces, left\_hands, right\_hands)$
$(gen\_hands, gen\_head) \leftarrow$ CropGeneratedRegions(gen_output)
$D\_real \leftarrow D(content, target)$
$D\_fake \leftarrow D(content, gen\_output)$
$D\_hand\_real \leftarrow D_{hand}(hand\_input, real\_hands)$
$D\_hand\_fake \leftarrow D_{hand}(hand\_input, gen\_hands)$
$D\_head\_real \leftarrow D_{head}(head\_input, real\_head)$
$D\_head\_fake \leftarrow D_{head}(head\_input, gen\_head)$
$L_{united} \leftarrow BCE(\text{avg}(D\_real, D\_hand\_real, D\_head\_real), 1) + BCE(\text{avg}(D\_fake, D\_hand\_fake, D\_head\_fake), 0)$
$L_D \leftarrow BCE(D\_real, 1) + BCE(D\_fake, 0) + \lambda_{Du} \cdot L_{united}$
$L_{D\_hand} \leftarrow BCE(D\_hand\_real, 1) + BCE(D\_hand\_fake, 0) + \lambda_{Du} \cdot L_{united}$
$L_{D\_head} \leftarrow BCE(D\_head\_real, 1) + BCE(D\_head\_fake, 0) + \lambda_{Du} \cdot L_{united}$
opt_D.zero_grad()
$opt_{D_{hand}}$.zero_grad()
$opt_{D_{head}}$.zero_grad()
L_D.backward()
L_{D_hand}.backward()
L_{D_head}.backward()
UpdateAllDiscriminators()
end procedure

Algorithm 4 Helper Procedures

procedure PROCESSKEYPOINTS(npy_list)
$bodies \leftarrow npy\_list[:, 0:23, :]$
$faces \leftarrow npy\_list[:, 23:91, :]$
$left\_hands \leftarrow npy\_list[:, 91:112, :]$
$right\_hands \leftarrow npy\_list[:, 112:, :]$
return $(bodies, faces, left\_hands, right\_hands)$
end procedure
procedure CROPREGIONS($content, target$)
Detect hand and head regions from keypoints
$hand\_input \leftarrow$ CropHandInput$(content)$
$real\_hands \leftarrow$ CropRealHands$(target)$
$head\_input \leftarrow$ CropHeadInput$(content)$
$real\_head \leftarrow$ CropRealHead$(target)$
return $(hand\_input, real\_hands, head\_input, real\_head)$
end procedure
procedure UPDATELEARNINGRATES
G.update_lr()
D.update_lr()
D_{hand}.update_lr()
D_{head}.update_lr()
Update part-specific learning rates
end procedure

Algorithm 5 Validation and Utility Procedures

```
procedure RUNVALIDATION
    Load validation batch
    Generate predictions
    Calculate metrics
    Save sample images
    Log to TensorBoard
end procedure
procedure SAVECHECKPOINTS
    Save generator weights
    Save discriminators weights
    Save optimizer states
end procedure
procedure UPDATEPARTPARAMETERS
    for each part (head, hand, global) do
        Calculate part-specific loss
        Backpropagate through part
        Update part parameters
    end for
end procedure
```