Semantic Line Diffusion: Character-Consistent Line Art from text-annotated Storyboards

Seo-Yeon Choi[1] and Kyungsu Lee[1]

Jeonbuk National University, Republic of Korea
Corresponding Author: Kyungsu Lee (ksl@jbnu.ac.kr)

Abstract. Generating clean line art from conti, rough storyboard sketches used in webtoon production, remains a labor-intensive process requiring substantial artistic expertise. Conti sketches provide only coarse geometry and sparse semantic cues, making it difficult to recover detailed line art while preserving consistent character identity across sequential panels. Existing sketch-to-image or diffusion-based translation models process images independently and therefore struggle to maintain cross-panel identity coherence. We present PanelDiff, a diffusion transformer framework for conti-to-line-art generation that explicitly models cross-panel character consistency. PanelDiff integrates three components. A multi-reference character conditioning module encodes character labels and multiple line-art exemplars into compact identity tokens for robust identity guidance under ambiguous conti inputs. A memory-augmented representation bank accumulates character-aware features from previously generated panels, while a similarity interpreter dynamically retrieves relevant entries for the current panel. A panel-aware diffusion transformer then jointly attends to conti structure, textual descriptions, identity tokens, and retrieved memory features to produce coherent line-art sequences. To support this task, we construct a conti–line-art dataset with character identity annotations and sequential panel structures. Experiments show that PanelDiff improves line-art fidelity, identity preservation, and cross-panel consistency over strong baselines, and ablation studies verify the contribution of each component. The code and dataset are available online (huggingface) *.

Keywords: Webtoon · Generative AI · Diffusion Models

1 Introduction

Recent advances in deep generative models have significantly improved the ability of artificial intelligence to synthesize realistic visual content across modalities such as text, images, audio, and video [9, 16]. In computer vision, diffusion-based and adversarial or autoencoder-based architectures have emerged as dominant paradigms for image generation [11,13,26]. Diffusion models generate high-quality images by reversing a progressive noising process [11], while latent diffu-

* SemanticDiffusion-for-Line-Art-code and SemanticDiffusion-for-Line-Art-dataset.

sion frameworks such as Stable Diffusion enable efficient text-conditioned synthesis in compressed latent spaces [26]. Diffusion Transformers (DiT) further improve scalability and multimodal conditioning by adopting transformer-based token representations [7,25]. In parallel, GANs and autoencoder-based approaches remain important for controllable or stylized generation [3,10,14,17]. Combined with vision–language grounding methods such as Grounding DINO [21], these frameworks enable spatially controllable multimodal image synthesis. Building on these advances, recent work has explored *visual storytelling*, which aims to generate coherent image sequences conditioned on narrative descriptions. Methods such as MagicScroll [28] and Intelligent Grimm [20] incorporate layout-aware diffusion and autoregressive reasoning for temporally consistent story generation, while StoryImager [27] and StoryDiffusion [35] address long-range narrative coherence and subject identity across image sequences. In parallel, sketch and line synthesis has been studied in single-image settings: LineArt Diffusion [30] focuses on structure-preserving line drawing, whereas SwiftSketch [2] and CLIPDraw++ [22] explore diffusion- and optimization-based vector sketch generation. Multimodal representation learning methods such as BLIP and Flamingo further align textual and visual representations for cross-modal reasoning [1,19].

Despite this progress, existing generative frameworks remain inadequate for structured visual storytelling tasks such as webtoon production (See Fig. 1). Webtoons are a widely adopted digital comic format produced through a structured workflow in which artists first create a rough storyboard called the *conti* and then refine each panel into clean line art. As a form of sequential visual narrative, comics and webtoons rely on panelized storytelling structures that organize images into coherent narrative sequences and guide narrative progression across panels [5]. In professional webtoon creation, the conti serves as a multimodal planning artifact that combines coarse sketches, textual annotations, and panel layouts to describe character actions, scene transitions, and spatial relationships [18]. Automatically transforming such multimodal storyboard inputs into polished line art therefore requires joint reasoning over visual and textual cues while preserving structural alignment and character identity across sequential panels. This task presents several challenges. First, conti sketches provide only sparse geometry and limited semantic cues, making accurate line-art re-

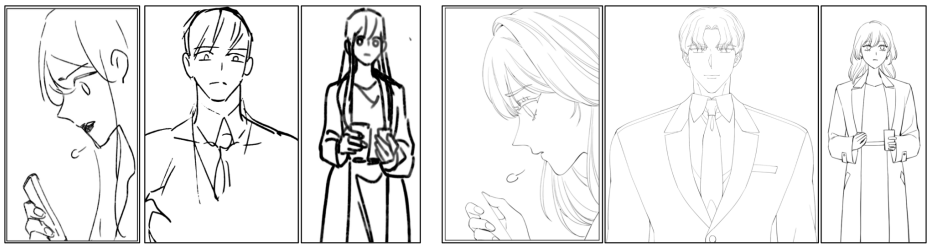

Fig. 1: Example of conti and corresponding line-art used in webtoon workflows.

construction difficult. Second, character identity must remain consistent across panels despite variations in pose, viewpoint, and expression, a common requirement in sequential visual narratives [5]. Third, webtoon narratives impose panel-level structural constraints such as vertical layout composition and discrete scene transitions. Existing sketch-to-image or text-to-image models typically generate images independently [25, 26], and therefore struggle to maintain cross-panel identity consistency and panel-level coherence throughout a sequence.

To address these challenges, we formulate the *conti-to-line art generation* problem, which aims to generate semantically faithful and structurally aligned line-art panel sequences from multimodal conti inputs. We propose **PanelDiff**, a diffusion transformer designed for storyboard-driven generation with explicit panel-structured reasoning and memory-based identity conditioning. PanelDiff introduces three key components: a multi-reference character conditioning module that encodes character labels and exemplar line art into compact identity tokens, a memory-augmented representation bank with similarity-based retrieval for maintaining cross-panel identity, and a panel-aware diffusion transformer that jointly attends to conti sketches, textual annotations, identity tokens, and retrieved memory features to produce coherent line-art sequences. Our contributions are summarized as follows:

- **Panel-Aware Multimodal Diffusion Framework.** We propose a diffusion transformer that jointly encodes conti sketches, textual annotations, and panel layout cues as structured tokens for conti-to-line art generation.
- **Memory-Augmented Identity Conditioning.** We introduce a retrieval-based identity conditioning module with multi-reference character embeddings and a memory representation bank to maintain character identity across non-contiguous panels.
- **Panel-Structured Webtoon Dataset and Evaluation.** We construct a conti–text–line dataset organized as character-aware panel sequences with identity annotations and panel ordering to evaluate structural alignment and cross-panel consistency (See Fig 2).

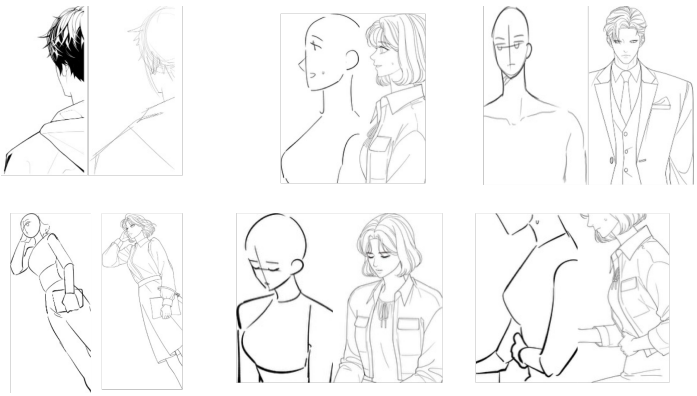

Fig. 2: Samples of proposed conti-line-art dataset.

2 Method

Our primary problem is to generate line-art panels conditioned on conti, textual character descriptions, multi-reference exemplars, and an evolving memory that enforces identity consistency across the sequence. We present the main formulation and model components below, and the extended derivations, additional equations, and detailed module architectures are deferred to the *Appendix*.

2.1 Problem Formulation

A webtoon episode is represented as a sequence of conti panels $\mathcal{P} = \{1, \ldots, T\}$, where each conti image $x_t \in \mathbb{R}^{H \times W \times 3}$ provides a rough layout and coarse depiction of characters. The objective is to generate the corresponding clean line-art image $y_t \in \mathbb{R}^{H \times W \times 3}$. Let $\mathcal{C}$ denote the global set of characters. For each character $c \in \mathcal{C}$, the artist provides a set of static identity reference images $\mathcal{R}_c = \{r_c^{(k)}\}$ that define the canonical appearance of the character and are independent of any episode or panel. Each conti panel t contains a subset of characters $\mathcal{C}_t \subseteq \mathcal{C}$, and their corresponding static references are always available.

To maintain identity consistency across panels, the model maintains a memory state $\mathcal{M}_t$ composed of two types of entries: static entries that store encoded features of reference images and dynamic entries that store character features extracted from previously generated line-art panels. The memory is defined as

$$\mathcal{M}_t = \{(\text{type}_j, c_j, f_j, \text{meta}_j) \mid j = 1, \ldots, M_t\}, \tag{1}$$

where $\text{type}_j \in \{\text{static}, \text{dynamic}\}$ indicates the entry type, $c_j \in \mathcal{C}$ is the associated character identity, $f_j \in \mathbb{R}^D$ is a character-level feature vector, and meta_j denotes auxiliary metadata for the entry. Particularly, meta_j includes the panel index, character bounding box, detector confidence, estimated pose/viewpoint, and associated text annotations when available. Static entries are created during initialization as

$$\mathcal{M}_0 = \text{Init}(\{\mathcal{R}_c\}_{c \in \mathcal{C}}), \tag{2}$$

while dynamic entries are appended according to the update rule

$$\mathcal{M}_t = U(\mathcal{M}_{t-1}, y_t). \tag{3}$$

Given the conti image x_t, textual character descriptions $\{l_c\}_{c \in \mathcal{C}_t}$, and the memory state $\mathcal{M}_{t-1}$, the model generates the line-art image as

$$y_t = G_\theta(x_t, \{l_c\}_{c \in \mathcal{C}_t}, \mathcal{M}_{t-1}). \tag{4}$$

We adopt a diffusion-based formulation by representing y_t with a latent variable $\mathbf{z}_t^{(0)}$ and applying the forward noising process

$$q(\mathbf{z}_t^{(s)} \mid \mathbf{z}_t^{(0)}) = \mathcal{N}(\alpha_s \mathbf{z}_t^{(0)}, \sigma_s^2 I). \tag{5}$$

The model learns a denoising function

$$\epsilon_\theta(\mathbf{z}_t^{(s)}, x_t, \{l_c\}_{c\in\mathcal{C}_t}, \mathcal{M}_{t-1}) \tag{6}$$

to approximate the reverse diffusion trajectory. This formulation enables the model to jointly leverage conti structure, textual descriptions, static identity priors, and dynamic identity history within a unified generative framework.

2.2 Overall Architecture

Given a conti panel x_t and its associated character metadata, our model generates the corresponding line-art image through a diffusion transformer conditioned on conti features, textual descriptions, and a unified memory bank that contains both static and dynamic identity information. Fig. 3 illustrates the detailed architecture of our framework. The conti image is encoded into a sequence of visual tokens $Z_t^x = E_x(x_t)$, and each textual description l_c is encoded into a text embedding e_c^{text} using a transformer encoder. All static reference images are processed once at initialization by a shared line-art encoder E_r to produce static identity features, which are stored as static entries in the memory state $\mathcal{M}_0$. After each generated panel y_t, the same encoder E_r extracts character-level features from detected regions of y_t, and these are appended as dynamic

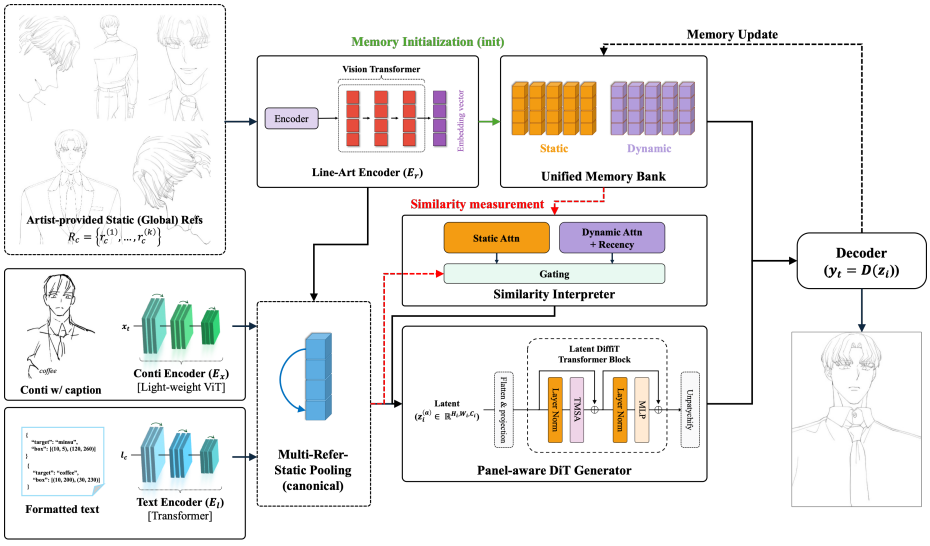

Fig. 3: Overview of our multimodal conti-to-line art framework. Artist-provided static references are encoded and stored in a unified memory bank together with dynamically accumulated identity features. Given the conti sketch and textual character descriptions, our model performs multi-reference static pooling and memory-based similarity interpretation to produce identity-aware character tokens. These, together with conti and text tokens, condition the panel-aware DiT generator to synthesize line-art panels, which are used to update the dynamic memory for cross-panel identity consistency.

entries to $\mathcal{M}_t$. Note that the conti encoder E_x is based on the light-weight vision transformer (ViT) [6], line-art encoder E_r is based on the hybrid construction of convolutional neural network (CNN) and ViT, and the text encoder E_l is based on the language transformer. The detailed descriptions are illustrated in Appendix.

For each character $c \in \mathcal{C}_t$, the similarity interpreter retrieves all memory entries associated with c and computes their relevance to the current conti context. This produces a memory-enhanced identity vector that is fused with the text embedding to form the final character token $\tilde{h}_{t,c}$. The conditioning sequence for the diffusion transformer is constructed by concatenating the panel index token p_t, conti tokens Z_t^x, text tokens, and the set $\{\tilde{h}_{t,c}\}$. At diffusion step s, the transformer predicts the noise component in the latent variable through:

$$\hat{\epsilon}_t^{(s)} = F_\theta(\mathbf{z}_t^{(s)}, Z_{t,s}^{\text{in}}, \gamma(s)), \tag{7}$$

where $Z_{t,s}^{\text{in}}$ denotes the assembled conditioning tokens and $\gamma(s)$ is the timestep embedding. After iterative denoising, the clean latent $\mathbf{z}_t^{(0)}$ is decoded into the final line-art output y_t, which subsequently updates the memory state. This architecture integrates static identity priors and episode-specific dynamic identity information to ensure fidelity to the conti and consistent character appearance across panels.

2.3 Multi-Reference Character Conditioning

Each character c appearing in panel t is associated with textual information l_c and static identity references stored in the memory state. The goal of the conditioning module is to form a character embedding that reflects both the canonical appearance defined by the static references and the semantic information provided by the text. Each reference image is encoded by the shared line-art encoder to produce feature vectors, and these features are retrieved as static entries from the unified memory $\mathcal{M}_{t-1}$. Let $F_{t,c}^{\text{static}}$ denote this set of static features. A context vector $u_{t,c}$ is derived from the conti tokens to represent the appearance implied by the current panel. The relevance of each static reference feature to the panel context is computed through attention weights

$$\alpha_{t,c,i} = \frac{\exp(u_{t,c}^\top W_r f_{t,c,i})}{\sum_j \exp(u_{t,c}^\top W_r f_{t,c,j})}, \tag{8}$$

where $f_{t,c,i} \in F_{t,c}^{\text{static}}$ and W_r is a learnable projection. The pooled static identity feature defined as below:

$$h_{t,c} = \sum_i \alpha_{t,c,i} f_{t,c,i}. \tag{9}$$

The pooled static reference representation is fused with the text embedding e_c^{text} to construct the initial character embedding that enters the similarity interpreter, which further integrates dynamic identity information from the memory. The result of this fusion becomes the character token supplied to the diffusion transformer.

2.4 Memory Bank

To ensure consistent character identity across panels, we maintain a unified memory state $\mathcal{M}_t$ containing both static identity information and dynamic appearance features accumulated during generation. The memory is initialized by encoding all artist-provided static reference images, which produce static feature vectors stored as static entries in $\mathcal{M}_0$. After generating each line-art panel y_t, character regions are extracted and encoded by the shared line-art encoder E_r to yield dynamic feature vectors, which are appended to the memory as dynamic entries. As defined in Eq. (1), each memory entry consists of its entry type, character index, character-level feature vector, and auxiliary metadata. The metadata term meta_j stores auxiliary information for each memory entry, including the panel index, character bounding box, detector confidence, estimated pose/viewpoint, and associated text annotations when available. For a character c appearing in panel t, the model retrieves from $\mathcal{M}_{t-1}$ all memory entries corresponding to c, including both static identity features and previously generated dynamic features. This unified set is passed to the similarity interpreter, allowing the model to adaptively combine global identity priors with episode-specific appearance cues when forming the character embedding used by the diffusion transformer. The update procedure for incorporating newly generated character features into the memory state is summarized in Algorithm 1.

Algorithm 1: Memory Update After Generating Panel t

Input: Previous memory $\mathcal{M}_{t-1}$; Generated line-art panel $\hat{y}_t$;
Line-art encoder E_r; Character regions $\{\Omega_{t,c}\}$;
Maximum dynamic memory size per character $K_{\text{mem}} = 64$
Output: Updated memory $\mathcal{M}_t$

Initialize:
$\mathcal{M}_t \leftarrow \mathcal{M}_{t-1}$; `// Start with previous memory`

Extract and update dynamic features:
foreach *character c appearing in panel t* **do**
  Extract region $\Omega_{t,c}$ from $\hat{y}_t$;
  Resize $\hat{y}_t|_{\Omega_{t,c}}$ to the input resolution of E_r;
  $f_{t,c}^{\text{dyn}} \leftarrow E_r(\hat{y}_t|_{\Omega_{t,c}})$;
  Generate metadata $\text{meta}_{t,c}$ containing panel index, bounding box, detector confidence, pose/viewpoint, and text annotations;
  `// Append new dynamic entry`
  $\mathcal{M}_t \leftarrow \mathcal{M}_t \cup \{(\text{dynamic}, c, f_{t,c}^{\text{dyn}}, \text{meta}_{t,c})\}$;
  `// Bound dynamic memory with a per-character FIFO buffer`
  $\mathcal{D}_{t,c} \leftarrow \{j \mid (\text{type}_j = \text{dynamic}) \wedge (c_j = c)\}$;
  if $|\mathcal{D}_{t,c}| > K_{\text{mem}}$ **then**
    Remove the oldest $|\mathcal{D}_{t,c}| - K_{\text{mem}}$ dynamic entries of character c from $\mathcal{M}_t$;

return $\mathcal{M}_t$

To prevent unbounded memory growth, dynamic entries are maintained using a bounded per-character FIFO buffer. Specifically, for each character, we keep at most $K_{\text{mem}} = 64$ dynamic entries. When the number of dynamic entries for a character exceeds this limit, the oldest entries are discarded. This keeps memory usage independent of the episode length while preserving recent character-specific appearance evidence.

2.5 Similarity Interpreter

For each character c in panel t, the similarity interpreter retrieves all memory features $F_{t,c} = \{f_{t,c,i}\}$ and decomposes them into static and dynamic subsets:

$$\mathcal{F}_{t,c}^{\text{static}} = \{f_{t,c,i} \mid \text{type}_i = \text{static}\}, \tag{10}$$

$$\mathcal{F}_{t,c}^{\text{dyn}} = \{f_{t,c,i} \mid \text{type}_i = \text{dynamic}\}. \tag{11}$$

Using this global conti representation and the text embedding, we construct a panel-adaptive query as:

$$u_{t,c} = W_u[g_t\,;\, e_c^{\text{text}}], \tag{12}$$

which captures the conti layout and semantic attributes of the current character depiction. Here, g_t denotes the global conti representation obtained by mean-pooling the conti token sequence Z_t^x followed by a linear projection. Subsequently, the static identity representations are aggregated via:

$$\alpha_{t,c,i}^{\text{static}} = \frac{\exp(u_{t,c}^\top W_{\text{st}} f_{t,c,i})}{\sum_{k \in \mathcal{F}_{t,c}^{\text{static}}} \exp(u_{t,c}^\top W_{\text{st}} f_{t,c,k})}, \tag{13}$$

$$s_{t,c} = \sum_{i \in \mathcal{F}_{t,c}^{\text{static}}} \alpha_{t,c,i}^{\text{static}} f_{t,c,i}. \tag{14}$$

For dynamic entries, we incorporate a temporal recency kernel:

$$\kappa(\Delta t_{t,i}) = \exp\left(-\frac{\Delta t_{t,i}}{\lambda_{\text{time}}}\right), \tag{15}$$

and aggregate the dynamic identity cues as:

$$\beta_{t,c,i}^{\text{dyn}} = \frac{\exp(u_{t,c}^\top W_{\text{dy}} f_{t,c,i})\,\kappa(\Delta t_{t,i})}{\sum_{k \in \mathcal{F}_{t,c}^{\text{dyn}}} \exp(u_{t,c}^\top W_{\text{dy}} f_{t,c,k})\,\kappa(\Delta t_{t,k})}, \tag{16}$$

$$d_{t,c} = \sum_{i \in \mathcal{F}_{t,c}^{\text{dyn}}} \beta_{t,c,i}^{\text{dyn}} f_{t,c,i}. \tag{17}$$

Static and dynamic identity cues are combined using a panel-adaptive gate:

$$\eta_{t,c} = \sigma\big(v^\top [g_t\,;\, e_c^{\text{text}}]\big)\,, \quad m_{t,c} = \eta_{t,c}\, s_{t,c} + (1 - \eta_{t,c})\, d_{t,c}. \tag{18}$$

Algorithm 2: Similarity Interpreter for Character c in Panel t

Input: Static memory features $\mathcal{F}_{t,c}^{\text{static}}$; Dynamic memory features $\mathcal{F}_{t,c}^{\text{dyn}}$;
Conti global feature g_t; Text embedding e_c^{text};
Canonical static embedding $\bar{h}_{t,c}$
Output: Identity token $\tilde{h}_{t,c}$

Query construction:
$u_{t,c} \leftarrow W_u[g_t\,;\, e_c^{\text{text}}]$;

Static memory attention:
foreach $f_{t,c,i} \in \mathcal{F}_{t,c}^{\text{static}}$ **do**
  $\alpha_{t,c,i}^{\text{static}} \leftarrow \exp(u_{t,c}^{\top} W_{\text{st}} f_{t,c,i})$;
Normalize $\alpha_{t,c,i}^{\text{static}}$ over all static entries;
$s_{t,c} \leftarrow \sum_i \alpha_{t,c,i}^{\text{static}} f_{t,c,i}$;

Dynamic memory attention with recency:
foreach $f_{t,c,i} \in \mathcal{F}_{t,c}^{\text{dyn}}$ **do**
  $\kappa(\Delta t_{t,i}) \leftarrow \exp(-\Delta t_{t,i}/\lambda_{\text{time}})$;
  $\beta_{t,c,i}^{\text{dyn}} \leftarrow \exp(u_{t,c}^{\top} W_{\text{dy}} f_{t,c,i}) \cdot \kappa(\Delta t_{t,i})$;
Normalize $\beta_{t,c,i}^{\text{dyn}}$ over all dynamic entries;
$d_{t,c} \leftarrow \sum_i \beta_{t,c,i}^{\text{dyn}} f_{t,c,i}$;

Panel-adaptive gating:
$\eta_{t,c} \leftarrow \sigma(v^{\top}[g_t\,;\, e_c^{\text{text}}])$;
$m_{t,c} \leftarrow \eta_{t,c} s_{t,c} + (1-\eta_{t,c}) d_{t,c}$;

Final identity embedding: $\tilde{h}_{t,c} \leftarrow \psi([\bar{h}_{t,c}\,;\, m_{t,c}])$;
return $\tilde{h}_{t,c}$;

This yields an identity representation tailored to the current panel. Finally, the memory-based identity $m_{t,c}$ is fused with the canonical static embedding $\bar{h}_{t,c}$ to produce the identity token supplied to the diffusion transformer:

$$\tilde{h}_{t,c} = \psi([\bar{h}_{t,c}\,;\, m_{t,c}]). \tag{19}$$

The complete retrieval–attention–gating procedure is detailed in Algorithm 2.

2.6 Panel Aware DiT Generator

The final line-art image is generated by a diffusion transformer that conditions on conti features, textual embeddings, and memory-enhanced character tokens. The input sequence for panel t is formed by concatenating the panel index token p_t, conti tokens Z_t^x, text tokens Z_t^{text}, and the character tokens $\{\tilde{h}_{t,c}\}$. These tokens provide structural cues, semantic information, and identity representations derived from both static and dynamic memory. At diffusion step s, the model receives a noised latent vector $\mathbf{z}_t^{(s)}$ and predicts the corresponding noise component using

$$\hat{\epsilon}_t^{(s)} = F_\theta(\mathbf{z}_t^{(s)}, Z_{t,s}^{\text{in}}, \gamma(s)), \tag{20}$$

where $Z_{t,s}^{\text{in}}$ denotes the assembled conditioning sequence and $\gamma(s)$ is the diffusion timestep embedding. Iterative denoising yields the clean latent $\mathbf{z}_t^{(0)}$, which is decoded into the line-art output y_t. The resulting image updates the memory state, enabling the model to maintain cross-panel identity consistency throughout the episode.

2.7 Training Losses

The model is trained using a diffusion-based denoising loss together with auxiliary objectives that enforce character identity preservation and cross-panel consistency. For each panel t, a latent representation $\mathbf{z}_t^{(0)}$ of the target line art undergoes the forward noising process to yield $\mathbf{z}_t^{(s)}$ at step s. The model predicts the noise component as $\hat{\epsilon}_t^{(s)} = F_\theta(\mathbf{z}_t^{(s)}, Z_{t,s}^{\text{in}}, \gamma(s))$, and the diffusion loss is defined as $L_{\text{diff}} = \mathbb{E}_{t,s}\left[\|\hat{\epsilon}_t^{(s)} - \epsilon_t^{(s)}\|_2^2\right]$. To maintain character identity, we use a feature-space contrastive loss that encourages features of the same character to remain close while separating those of different characters. Let $f_{t,c}$ and $f_{t',c}$ denote memory features of the same character and let $f_{t,c}^{\text{neg}}$ denote a feature of a different character. The identity loss is defined as $L_{\text{id}} = \mathbb{E}\big[\max\big(0, \|f_{t,c} - f_{t',c}\|_2 - \|f_{t,c} - f_{t,c}^{\text{neg}}\|_2 + \delta\big)\big]$. To stabilize identity across adjacent panels, we introduce a temporal consistency loss that penalizes deviations between dynamic memory features extracted from consecutive panels:

$$L_{\text{cons}} = \mathbb{E}_{c,t}[\|f_{t,c} - f_{t-1,c}\|_2]\,. \tag{21}$$

The overall objective is $L = L_{\text{diff}} + \lambda_{\text{id}} L_{\text{id}} + \lambda_{\text{cons}} L_{\text{cons}}$.

3 Experiments

We construct our custom conti–line-art dataset because no public benchmark exists for this task. The dataset contains about 100k paired panels created by 30 professional webtoon artists and includes roughly 150 recurring characters with static reference images. We use three splits: a standard split for main evaluation, an unseen-character split in which selected characters are held out during training, and an unseen-artist split holding out complete works of specific artists. To validate our framework, we compare our method with CycleGAN [36], Pix2PixHD [29], SPADE [24], Stable Diffusion [26], ControlNet [33], T2I-Adapter [23], IP-Adapter [32], DiffSketcher [31], InstructPix2Pix [4], and GPT-4o [12]. All baselines are trained or fine-tuned on the standard training split at 512×512 resolution with identical preprocessing. Our model is trained with a diffusion transformer backbone using a cosine noise schedule, AdamW optimizer, batch size 64, and precomputed static reference embeddings. All models are trained for 100 epochs using the AdamW optimizer [15]. The initial learning rate is set to 10^{-3} and is reduced to 10^{-5} when convergence becomes unstable. All experiments are conducted in the same environment: each device is equipped with eight NVIDIA RTX A6000 GPUs and 128GB RAM.

3.1 Comparison Analysis

We evaluate single-panel conti-to-line art generation by comparing our method with baseline image-to-image and diffusion-based models trained under identical conditions. For each conti panel, models predict a corresponding line-art image, and we measure LPIPS [34], FID computed on line-art encoder features [8], MS-SSIM computed on automatically detected character-face regions, PSNR for reconstruction fidelity, and Edge IoU to evaluate structural alignment of generated line strokes. As shown in Table 1, baselines struggle to preserve fine identity details, often producing distorted facial geometry or inconsistent stroke patterns even when overall layout is correct. Diffusion-based approaches improve global coherence but remain limited in capturing detailed, character-specific line structure. Our method achieves the lowest LPIPS and FID and the highest face-region MS-SSIM, indicating stronger preservation of identity-critical geometry and more faithful alignment with the underlying conti.

Table 1: Single-panel conti-to-line-art generation performance on our Webtoon Conti–LineArt dataset. We compare PanelDiff with image-to-image translation, sketch-conditioned generation, reference-conditioned generation, and multimodal generative baselines. Metrics evaluate perceptual fidelity (LPIPS, FID, MS-SSIM, PSNR, SSIM), structural alignment (Edge IoU, Edge F1), and character identity preservation (ID Sim). We report the mean over five random seeds, with standard deviations shown in parentheses.

Model	Fidelity				Structure			Identity
	LPIPS ↓	FID ↓	MS-SSIM ↑	PSNR ↑	SSIM ↑	Edge IoU ↑	Edge F1 ↑	ID Sim ↑
CycleGAN	0.345 (0.021)	91.7 (2.40)	0.812 (0.018)	18.3 (0.03)	0.731 (0.020)	0.54 (0.02)	0.58 (0.02)	0.71 (0.02)
Pix2PixHD	0.312 (0.019)	78.4 (2.10)	0.835 (0.017)	19.7 (0.02)	0.752 (0.019)	0.58 (0.02)	0.61 (0.02)	0.74 (0.02)
SPADE	0.287 (0.018)	72.1 (1.90)	0.842 (0.016)	20.6 (0.02)	0.768 (0.018)	0.60 (0.02)	0.64 (0.02)	0.76 (0.02)
Stable Diffusion	0.258 (0.016)	61.3 (1.70)	0.857 (0.015)	21.9 (0.02)	0.781 (0.017)	0.64 (0.02)	0.67 (0.02)	0.79 (0.02)
ControlNet	0.233 (0.015)	55.1 (1.50)	0.864 (0.014)	22.6 (0.02)	0.792 (0.016)	0.66 (0.02)	0.69 (0.02)	0.81 (0.02)
T2I-Adapter	0.225 (0.014)	53.2 (1.40)	0.868 (0.014)	23.1 (0.02)	0.799 (0.015)	0.67 (0.02)	0.70 (0.02)	0.82 (0.02)
IP-Adapter	0.216 (0.014)	51.4 (1.30)	0.874 (0.013)	23.6 (0.02)	0.807 (0.015)	0.68 (0.02)	0.72 (0.02)	0.83 (0.02)
DiffSketcher	0.208 (0.013)	49.8 (1.20)	0.881 (0.012)	24.0 (0.02)	0.815 (0.014)	0.69 (0.02)	0.73 (0.02)	0.84 (0.02)
InstructPix2Pix	0.241 (0.016)	64.4 (1.80)	0.853 (0.015)	21.2 (0.03)	0.776 (0.017)	0.63 (0.02)	0.66 (0.02)	0.78 (0.02)
GPT-4o	0.274 (0.017)	69.2 (2.00)	0.846 (0.016)	20.4 (0.03)	0.763 (0.018)	0.61 (0.02)	0.64 (0.02)	0.76 (0.02)
Ours (full)	**0.162** (0.011)	**39.8** (0.90)	**0.903** (0.010)	**26.8** (0.01)	**0.861** (0.011)	**0.74** (0.01)	**0.80** (0.01)	**0.91** (0.01)

3.2 Episode-Level Consistency on Sequences

We assess long-range character and structural stability using sequences of 50 consecutive conti panels. While baseline methods generate each panel independently, our model maintains a memory-based identity representation that evolves across the sequence. Identity Drift is defined as: $\text{Drift}(t, c) = \left\| f_{t,c}^{\text{dyn}} - f_{t+1,c}^{\text{dyn}} \right\|_2$, and we also measure a sequence-level identity similarity score and layout consistency computed as the IoU between conti bounding regions and the corresponding regions estimated from generated line art. As shown in Figure 4 and Table 2, baseline methods accumulate drift over the sequence, causing noticeable degradation

Fig. 4: Identity drift over 50-cut sequences. Baselines exhibit rapid drift, whereas our method maintains nearly constant identity.

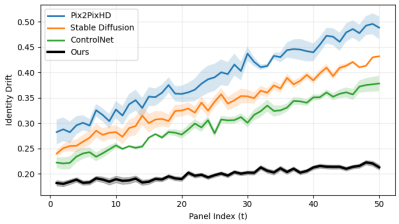

Table 2: Sequence-level consistency and computational cost. Lower is better for Drift, higher is better for MS-SSIM and Layout IoU. FLOPs are estimated for a single 512×512 panel generation.

Model	Drift ↓	MS-SSIM ↑	Layout IoU ↑	FLOPs (G)
CycleGAN [36]	0.482 ± 0.021	0.33 ± 0.02	0.41 ± 0.01	23
Pix2PixHD [29]	0.451 ± 0.018	0.37 ± 0.02	0.45 ± 0.01	69
SPADE [24]	0.429 ± 0.017	0.40 ± 0.02	0.48 ± 0.01	92
Stable Diffusion [26]	0.377 ± 0.014	0.44 ± 0.02	0.52 ± 0.01	150
ControlNet [33]	0.355 ± 0.013	0.47 ± 0.02	0.56 ± 0.01	188
T2I-Adapter [23]	0.346 ± 0.012	0.49 ± 0.01	0.58 ± 0.01	152
IP-Adapter [32]	0.337 ± 0.012	0.51 ± 0.01	0.59 ± 0.01	149
DiffSketcher [31]	0.328 ± 0.011	0.52 ± 0.01	0.60 ± 0.01	34
InstructPix2Pix [4]	0.392 ± 0.016	0.43 ± 0.02	0.50 ± 0.01	160
GPT-4o [12]	0.371 ± 0.015	0.45 ± 0.02	0.53 ± 0.01	—
Ours (full)	**0.214 ± 0.009**	**0.68 ± 0.01**	**0.71 ± 0.01**	**41**

in facial geometry and other identity-sensitive details. In contrast, our model maintains significantly lower drift, higher identity similarity, and more stable layout alignment across all 50 panels. These results indicate that long-range consistency cannot be achieved by per-panel generation alone and demonstrate the benefit of memory-based identity modeling.

3.3 Ablation Studies

We analyze the contribution of each core component of our framework by removing (1) multi-reference conditioning, (2) the memory bank, and (3) the similarity interpreter. As shown in Table 3, removing multi-reference conditioning noticeably reduces MS-SSIM and increases identity drift, indicating that diverse static exemplars provide essential canonical cues. Using only a single reference improves stability but still underperforms compared to the full variant. Eliminating the memory bank yields the largest drift and the lowest structural fidelity, whereas restricting it to the most recent panel partially mitigates the issue but accumulates errors over longer sequences. Removing the similarity interpreter also degrades performance, demonstrating the importance of adaptive weighting be-

Table 3: Ablation study on 50-cut sequences. We evaluate the contribution of multi-reference conditioning, dynamic memory, and the similarity interpreter using identity preservation, temporal consistency, and structural alignment metrics. Values denote the mean over five random seeds, with standard deviations shown in parentheses.

Variant	MS-SSIM ↑	Drift ↓	Layout IoU ↑	ID Sim ↑	Edge IoU ↑
Ours (full)	**0.903** (0.002)	**0.214** (0.009)	**0.71** (0.01)	**0.91** (0.01)	**0.74** (0.01)
w/o Multi-Reference	0.861 (0.003)	0.301 (0.012)	0.65 (0.01)	0.84 (0.01)	0.67 (0.01)
Single Reference	0.878 (0.003)	0.268 (0.011)	0.67 (0.01)	0.86 (0.01)	0.69 (0.01)
w/o Memory Bank	0.842 (0.004)	0.332 (0.015)	0.62 (0.01)	0.82 (0.01)	0.64 (0.01)
Recent-Only Memory	0.874 (0.003)	0.259 (0.010)	0.66 (0.01)	0.87 (0.01)	0.69 (0.01)
w/o Similarity Interpreter	0.851 (0.004)	0.317 (0.013)	0.63 (0.01)	0.83 (0.01)	0.65 (0.01)

tween static and dynamic cues. Overall, all components contribute meaningfully to structural stability and the preservation of long-range identity.

3.4 Generalization to Unseen Scenario

We further evaluate whether our model generalizes to unseen characters and unseen artistic styles, which is essential for practical webtoon production pipelines. We construct two evaluation splits: (1) unseen-character episodes containing characters never observed during training, and (2) unseen-artist episodes composed entirely of panels drawn by artists excluded from the training set. All baselines receive the same static reference exemplars as our model. As shown in Table 4, most baselines show substantial degradation in both identity stability and structural alignment under these domain shifts. DiffSketcher and T2I-Adapter maintain sharper strokes but fail to adapt consistently to unseen styles. In contrast, our model achieves the highest MS-SSIM and layout alignment while maintaining lower drift, demonstrating that multi-reference conditioning and memory-based aggregation enable robust generalization beyond the training distribution.

Table 4: Generalization to unseen characters and artists under identity and structural alignment metrics.

Model	Unseen Characters					Unseen Artists				
	MS-SSIM ↑	Drift ↓	IoU ↑	ID Sim ↑	Edge IoU ↑	MS-SSIM ↑	Drift ↓	IoU ↑	ID Sim ↑	Edge IoU ↑
CycleGAN	0.645	0.462	0.34	0.63	0.38	0.632	0.478	0.32	0.61	0.36
Pix2PixHD	0.668	0.431	0.37	0.66	0.41	0.658	0.445	0.35	0.64	0.39
SPADE	0.682	0.417	0.39	0.67	0.43	0.671	0.429	0.37	0.65	0.41
Stable Diffusion	0.695	0.392	0.42	0.69	0.46	0.683	0.407	0.40	0.67	0.44
ControlNet	0.701	0.381	0.45	0.71	0.48	0.690	0.395	0.43	0.69	0.46
T2I-Adapter	0.708	0.374	0.47	0.72	0.49	0.695	0.388	0.45	0.70	0.47
IP-Adapter	0.712	0.363	0.48	0.73	0.50	0.699	0.376	0.46	0.71	0.48
DiffSketcher	0.715	0.352	0.49	0.74	0.51	0.702	0.365	0.47	0.72	0.49
InstructPix2Pix	0.682	0.408	0.41	0.68	0.44	0.670	0.418	0.38	0.66	0.42
GPT-4o	0.676	0.397	0.44	0.69	0.46	0.665	0.405	0.42	0.67	0.44
Ours (full)	**0.803**	**0.271**	**0.58**	**0.87**	**0.64**	**0.792**	**0.284**	**0.56**	**0.85**	**0.62**

3.5 Robustness and Reference Conditioning

In practical webtoon production, conti panels are often noisy or partially incomplete. To evaluate robustness, we apply controlled perturbations to the conti input, including line deletion, stroke jitter, and bounding-region shifts. Performance is measured using face-region MS-SSIM and layout IoU. For a character c in panel t, layout IoU is computed between the conti-provided bounding region $B_{t,c}^{conti}$ and the corresponding region estimated from generated line art $B_{t,c}^{gen}$ as $\mathrm{IoU}(t,c) = \frac{|B_{t,c}^{conti} \cap B_{t,c}^{gen}|}{|B_{t,c}^{conti} \cup B_{t,c}^{gen}|}$. Panel-level layout accuracy is obtained by averaging IoU across characters. As shown in Table 5, baseline performance drops sharply

Table 5: Robustness to perturbation.

Noise Level	Baseline Avg.		Ours	
	MS-SSIM	IoU	MS-SSIM	IoU
0.0	0.82	0.52	0.90	0.71
0.25	0.74	0.45	0.87	0.68
0.50	0.63	0.38	0.84	0.65
1.00	0.48	0.30	0.79	0.59

Table 6: Performance as a function of the number of static references per character.

K references	MS-SSIM ↑	Drift ↓	Layout IoU ↑
1	0.78	0.32	0.55
2	0.82	0.30	0.57
4	0.86	0.27	0.59
8	0.88	0.26	0.60

even under mild perturbations, whereas our method maintains higher MS-SSIM and IoU across all noise levels, indicating stronger robustness to imperfect conti inputs. We further analyze the effect of the number of static reference images used for identity conditioning. For each character, we sample $K \in \{1, 2, 4, 8\}$ reference images and evaluate face-region MS-SSIM, identity drift, and layout IoU. As shown in Table 6, performance improves consistently as more references are provided, with the largest gains observed between $K = 1$ and $K = 4$. This result suggests that diverse reference exemplars provide richer identity cues and that our conditioning module effectively aggregates information.

The additional experiments (partially in Fig 5) are in Appendix.

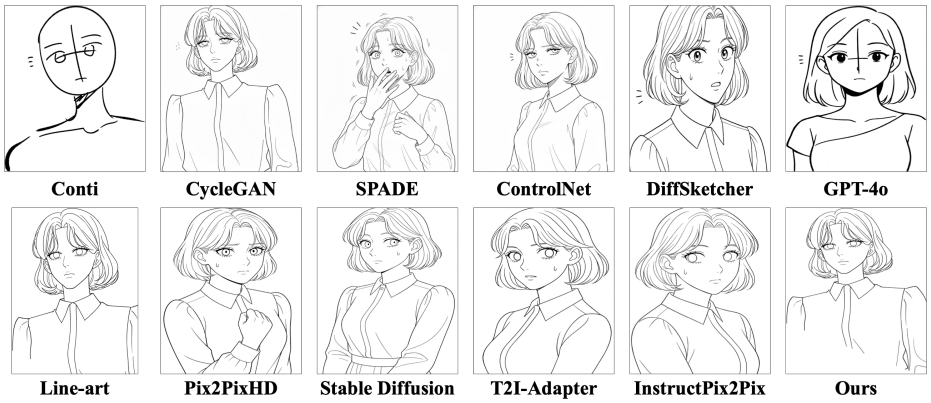

Fig. 5: Qualitative analysis of conti-to-line art synthesis.

4 Discussion

To the best of our knowledge, no prior work has directly addressed conti-to-line art generation or explored AI pipelines for webtoon production. While related areas such as visual storytelling and sketch-conditioned generation have been studied, existing models and datasets fail to capture the multimodal structure and sequential constraints of professional conti workflows. Our framework, therefore,

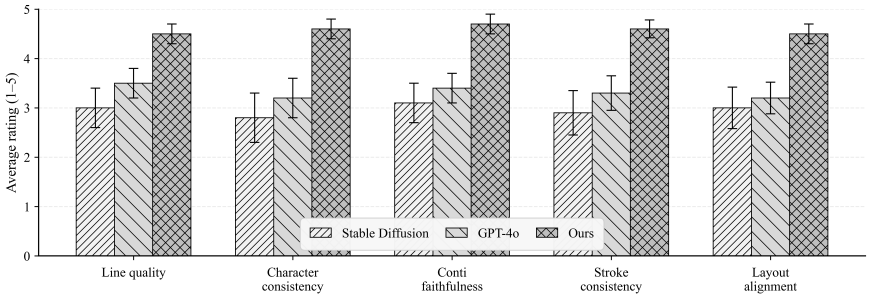

Fig. 6: User study results from thirty professional webtoon artists comparing Stable Diffusion, GPT-4o, and our method.

represents an initial step toward formalizing this task and establishing a generative paradigm for structured sequential illustration, supported by our dataset. In addition to quantitative evaluation, we conducted a user study (See Fig. 6) with thirty professional webtoon artists, none of whom contributed to the training dataset, comparing Stable Diffusion, GPT-4o, and our method in terms of line quality, character consistency, and conti alignment. Our model was consistently preferred across all criteria, indicating stronger preservation of structural cues and identity traits required in real production workflows. *Detailed protocols and rating statistics are provided in the appendix.*

5 Conclusion

In this research, we present a generative framework for conti-to-line art synthesis tailored to the structural and narrative requirements of webtoon creation. Our framework integrates multimodal conti inputs, static reference images, and dynamically accumulated identity features through a unified memory bank. The proposed similarity interpreter enables panel-adaptive retrieval and fusion of static and dynamic identity cues, while a panel-aware DiT generator incorporates structural, semantic, and identity-conditioned signals to produce coherent line-art panels. We also introduce a new conti–text–line art dataset derived from real-world webtoon workflows, providing a benchmark for storyboard-conditioned generation. Extensive experiments show that our approach significantly improves character consistency, semantic alignment, and structural fidelity compared to existing baselines. We hope this work provides a foundation for scalable webtoon automation and encourages further research in multimodal storyboard-conditioned visual synthesis.

6 Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-02263810)

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems **35**, 23716–23736 (2022) 2
2. Arar, E., Frenkel, Y., Cohen-Or, D., Shamir, A., Vinker, Y.: Swiftsketch: A diffusion model for image-to-vector sketch generation. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. pp. 1–12 (2025) 2
3. Bank, D., Koenigstein, N., Giryes, R.: Autoencoders (2021), https://arxiv.org/abs/2003.05991 2
4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18392–18402 (2023) 10, 12
5. Cohn, N.: The Visual Language of Comics: Introduction to the Structure and Cognition of Sequential Images. A&C Black (2013) 2, 3
6. Dosovitskiy, A.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 6
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale (2021), https://arxiv.org/abs/2010.11929 2
8. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024) 11
9. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014) 1
10. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks (2014), https://arxiv.org/abs/1406.2661 2
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 1
12. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024) 10, 12
13. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019) 1
14. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks (2019), https://arxiv.org/abs/1812.04948 2
15. Kingma, D.P.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 10
16. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013) 1
17. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2022), https://arxiv.org/abs/1312.6114 2

18. Ko, H.K., An, S., Park, G., Kim, S.K., Kim, D., Kim, B., Jo, J., Seo, J.: Wetoon: A communication support system between writers and artists in collaborative webtoon sketch revision. In: Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology. pp. 1–14 (2022) 2
19. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International conference on machine learning. pp. 12888–12900. PMLR (2022) 2
20. Liu, C., Wu, H., Zhong, Y., Zhang, X., Wang, Y., Xie, W.: Intelligent grimm-open-ended visual storytelling via latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6190–6200 (2024) 2
21. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., Zhang, L.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection (2024), https://arxiv.org/abs/2303.05499 2
22. Mathur, N., Marjit, S., Chaudhuri, A., Dutta, A.: Clipdraw++: Text-to-sketch synthesis with simple primitives. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6247–6256 (2025) 2
23. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In: Proceedings of the AAAI conference on artificial intelligence. vol. 38, pp. 4296–4304 (2024) 10, 12
24. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2337–2346 (2019) 10, 12
25. Peebles, W., Xie, S.: Scalable diffusion models with transformers (2023), https://arxiv.org/abs/2212.09748 2, 3
26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 1, 2, 3, 10, 12
27. Tao, M., Bao, B.K., Tang, H., Wang, Y., Xu, C.: Storyimager: A unified and efficient framework for coherent story visualization and completion. In: European Conference on Computer Vision. pp. 479–495. Springer (2024) 2
28. Wang, B., Meng, H., Cai, Z., Li, L., Ma, Y., Chen, Q., Wang, Z.: Magicscroll: Nontypical aspect-ratio image generation for visual storytelling via multi-layered semantic-aware denoising. arXiv preprint arXiv:2312.10899 (2023) 2
29. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018) 10, 12
30. Wang, X., Li, H., Fang, H., Peng, Y., Xie, H., Yang, X., Li, C.: Lineart: A knowledge-guided training-free high-quality appearance transfer for design drawing with diffusion model. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2912–2923 (2025) 2
31. Xing, X., Wang, C., Zhou, H., Zhang, J., Yu, Q., Xu, D.: Diffsketcher: Text guided vector sketch synthesis through latent diffusion models. Advances in Neural Information Processing Systems **36**, 15869–15889 (2023) 10, 12
32. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721 (2023) 10, 12

33. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3836–3847 (2023) 10, 12
34. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018) 11
35. Zhou, Y., Zhou, D., Cheng, M.M., Feng, J., Hou, Q.: Storydiffusion: Consistent self-attention for long-range image and video generation. Advances in Neural Information Processing Systems **37**, 110315–110340 (2024) 2
36. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017) 10, 12