

REAL-TIME MOTION-CONTROLLABLE AUTOREGRESSIVE VIDEO DIFFUSION

000
001
002
003
004
005 **Anonymous authors**
006 Paper under double-blind review
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053
054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
190

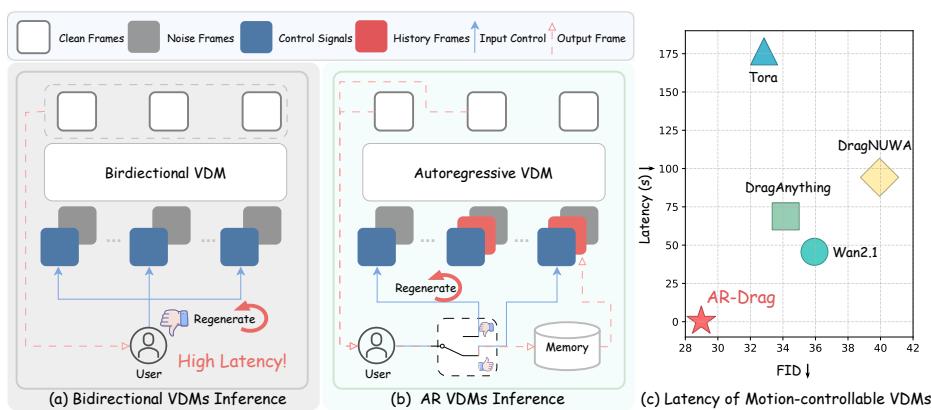


Figure 1: Comparison for motion-controllable video generation. (a) Bidirectional VDMs denoise all frames jointly; motion control can be adjusted only after all frames are generated, causing high latency. (b) In contrast, AR VDMs generate frames sequentially; motion control can be updated frame by frame and, if unsatisfactory, regenerated on the fly, enabling real-time adjustment. (c) Our method achieves significantly lower latency while maintaining superior FID performance.

condition on ground-truth frames during training rather than self-generated ones, breaking the MDP formulation; (2) handling the long decision process of video generation, where exploration across the entire decision chain becomes prohibitively expensive; (3) the lack of well-defined reward models tailored to controllable video generation.

To address these issues, we propose AR-Drag, an RL-enhanced few-step AR VDM for real-time motion-controllable I2V generation. Specifically, we first fine-tune the Wan2.1-1.3B (Wan et al., 2025) I2V model on our curated control-aware data to enable basic motion control, and then further improve it through reinforcement learning. To preserve the Markov property, we introduce **Self-Rollout**, training on model-generated histories to align with AR inference. To keep long-horizon exploration tractable, we adopt **selective stochasticity**: a single randomly chosen denoising step uses an SDE update, while all remaining steps follow the deterministic ODE solver. In addition, we design a trajectory-based reward model to enforce fine-grained control over complex motion signals.

Our contributions are threefold: (1) We propose AR-Drag, the first few-step AR VDM capable of real-time controllable I2V generation. (2) We introduce RL-based training for AR VDM and design a trajectory-based reward model tailored to fine-grained motion alignment. (3) We conduct extensive experiments showing that AR-Drag significantly improves both visual quality and controllability, despite using only 1.3B parameters.

2 RELATED WORKS

Controllable video generation. Early methods (Jeong et al., 2024; Wang et al., 2023; Zhao et al., 2024) achieve motion control by injecting motion signals into VDMs, yet their capability is restricted to reproducing pre-defined dynamics. Recent works (Geng et al., 2025; Ma et al., 2024; Mou et al., 2024; Shi et al., 2024; Wang et al., 2024; Yin et al., 2023; Zhang et al., 2025; Wu et al., 2024; Wang et al., 2025) leverage explicit control inputs such as motion trajectories, offering greater flexibility. For example, DragNUWA (Yin et al., 2023) conditions on trajectories to model camera and object motions, DragAnything (Wu et al., 2024) leverages object masks for entity-level control, and Tora (Zhang et al., 2025) introduces trajectory conditioning into a DiT framework. However, all these methods are non-autoregressive and therefore unsuitable for real-time interactive control.

Real-time video generation. Video diffusion models typically adopt bidirectional attention mechanism (Blattmann et al., 2023a;b; Brooks et al., 2024; Ho et al., 2022; Kong et al., 2024; Villegas et al., 2022; Wan et al., 2025; Yang et al., 2024). While effective for quality, this design requires jointly denoising all frames of video, limiting their applicability to real-time interactive. Autoregressive models (Hu et al., 2024; Jin et al., 2024; Yin et al., 2025; Gao et al., 2024; Gu et al., 2025; Li et al., 2025b), in contrast, generate tokens sequentially, making them inherently better suited for real-time controllable video generation. Some attempts (Yin et al., 2025; Lin et al., 2025; Yang

108 et al., 2025) distill multi-step VDMs into few-step autoregressive VDMs using distribution matching
 109 distillation (Yin et al., 2024b;a) or consistency distillation (Song et al., 2023; Song & Dhariwal,
 110 2023). However, AR VDMs still exhibit a train–test mismatch, making them prone to error
 111 accumulation across frames—particularly in few-step models. To mitigate this, some works (Chen
 112 et al., 2024; Teng et al., 2025; Sun et al., 2025) propose progressive noise schedules that gradu-
 113 ally increase noise from early to later frames, partially alleviating error accumulation. However,
 114 they neither close the train–test gap nor support real-time interaction, since future frames must be
 115 pre-generated before the current frame is rendered, introducing latency and limiting control effec-
 116 tiveness. Self-Forcing (Huang et al., 2025) narrows the train–test gap and improves stability by
 117 unrolling autoregressive generation during training, conditioning each frame on previously gener-
 118 ated outputs rather than ground truth. However, it does not strictly follow the autoregressive chain
 119 rule and leaves residual discrepancies (see Sec. 3.2). In contrast, our Self-Rollout strategy strictly
 120 adheres to the chain rule and aligns training with inference, providing a more principled formulation
 121 for integration with reinforcement learning.

122 **Alignment for diffusion model.** Existing approaches include scalar reward fine-tuning (Prabhude-
 123 sain et al., 2023; Clark et al., 2023; Xu et al., 2023; Prabhudesai et al., 2024), Reward-Weighted
 124 Regression (RWR) (Peng et al., 2019; Lee et al., 2023; Furuta et al., 2024), and Direct Preference
 125 Optimization (DPO)-based methods (Rafailov et al., 2023; Wallace et al., 2024; Dong et al., 2023).
 126 However, policy gradient methods (Schulman et al., 2017; Fan et al., 2023) often suffer from insta-
 127 bility. To improve stability in generative modeling, recent works such as DanceGRPO (Xue et al.,
 128 2025) and FlowGRPO (Liu et al., 2025) extend GRPO to flow-matching models. Building on this
 129 line of research, we extend GRPO to the I2V setting, achieving improved motion controllability
 130 while maintaining visual quality and efficiency.

131 3 METHOD

132 Our AR-Drag has two steps: In step 1 (Section 3.2), we build a real-time AR base model with
 133 basic motion control ability—assemble control-aware data, train a bidirectional teacher, and distill
 134 to a few-step causal student; during distillation we introduce Self-Rollout to align training with
 135 AR inference. In step 2 (Section 3.3), we treat AR video generation as an MDP and optimize with
 136 GRPO, designing selective stochastic sampling and a reward to improve realism and motion control.
 137

138 3.1 PRELIMINARY

139 **Flow matching.** Given a prior $p_0(\mathbf{x})$ and target data distribution $p_1(\mathbf{x})$, flow matching constructs an
 140 interpolating distribution $p_t(\mathbf{x})$. The sample trajectory \mathbf{x}_t follows the probability flow ODE:

$$141 \frac{d\mathbf{x}_t}{dt} = \mathbf{v}_\theta(\mathbf{x}_t, t), \quad \mathbf{x}_0 \sim p_0. \quad (1)$$

142 The training objective minimizes the squared error between the predicted vector field \mathbf{v}_θ and the
 143 ground-truth flow \mathbf{v} :

$$144 \mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, \mathbf{x}_t} [\|\mathbf{v}_\theta(\mathbf{x}_t, t) - \mathbf{v}\|_2^2], \quad (2)$$

145 where the target velocity field is $\mathbf{v} = \mathbf{x}_1 - \mathbf{x}_0$.

146 **Flow-ODE to SDE.** In flow-based probability models, the forward process is deterministic and
 147 follows an ODE: $d\mathbf{x}_t = \mathbf{v}_t dt$. To introduce stochasticity while preserving the same marginal distri-
 148 butions, a reverse-time SDE formulation can be defined as:

$$149 d\mathbf{x}_t = (\mathbf{v}_t(\mathbf{x}_t) - \frac{1}{2}\sigma_t^2 \nabla \log p_t(\mathbf{x}_t)) dt + \sigma_t d\mathbf{w}, \quad (3)$$

150 which leads to the update rule:

$$151 \mathbf{x}_{t+\Delta t} = \mathbf{x}_t + [\mathbf{v}_\theta(\mathbf{x}_t, t) + \frac{1}{2t}\sigma_t^2(\mathbf{x}_t + (1-t)\mathbf{v}_\theta(\mathbf{x}_t, t))] \Delta t + \sigma_t \sqrt{\Delta t} \epsilon. \quad (4)$$

152 **Distribution matching distillation (DMD).** DMD distills a multi-step teacher model into a few-step
 153 student model (Yin et al., 2024b;a) by minimizing the KL divergence between student-generated
 154 distribution $p_{\theta, t}$ and data distribution distribution $p_{\text{data}, t}$ across randomly sampled time t :

$$155 \mathbb{E}_t [\text{KL}(p_{\theta, t} || p_{\text{data}, t})] \quad (5)$$

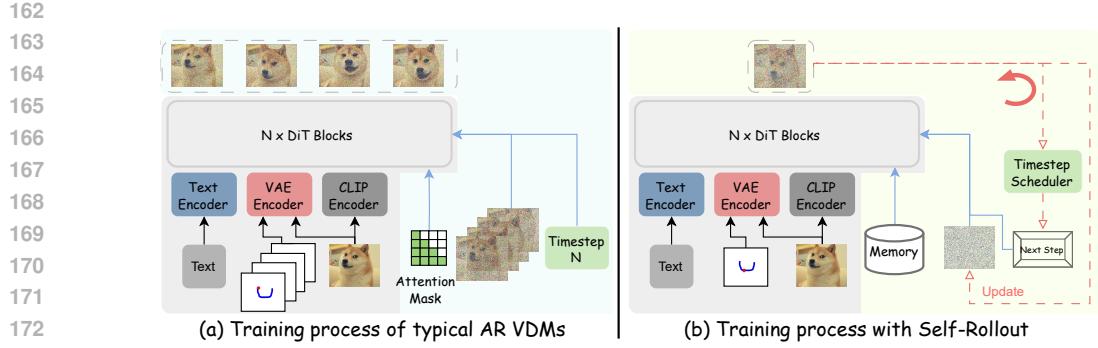


Figure 2: Comparison between typical AR VDMs and Self-Rollout. Self-Rollout faithfully follows the inference process during training, minimizing the train–test gap and naturally preserving the Markov property.

3.2 STEP 1: FINE-TUNING A REAL-TIME MOTION-CONTROLLABLE BASE VDM

In Step 1, we build a base AR VDM with basic real-time motion control by (i) curating videos with control signals, (ii) fine-tuning a bidirectional VDM on this data to learn motion control, (iii) distilling it into a few-step causal AR model for real-time inference with Self-Rollout, which “Markovize” AR training and paves the way for GRPO in Step 2.

Data curation. We collect a training corpus of real and synthetic videos featuring diverse motions. Control signals are obtained by generating keypoint trajectories with an automatic detector (Doersch et al., 2022) and retaining only samples that pass human verification. For challenging cases, such as occlusion or fast motion, we additionally curate a high-quality dataset that is fully annotated by human annotators. Our curated corpus encompasses a rich spectrum of actions and visual styles—spanning humans, animals, and cartoons—and includes videos of varying resolutions and durations, making it well-suited for evaluating generalization across diverse scenarios. In addition, each video is accompanied by rich textual descriptions (both positive and negative prompts). Please refer to Appendix B.1 for the details.

Bidirectional fine-tuning with motion-control. At m -th frame, we use three control signals

$$c_m = \begin{cases} (c_m^{\text{traj}}, c^{\text{text}}, c^{\text{ref}}), & m = 0, \\ (c_m^{\text{traj}}, c^{\text{text}}, \emptyset), & \text{otherwise.} \end{cases} \quad (6)$$

Here, c_m^{traj} is a motion-trajectory embedding obtained by encoding the raw coordinate heatmap at frame m with a VAE encoder (Wan et al., 2025). c^{text} encodes the textual signal, combining both positive and negative prompts. The text embedding is shared across all frames. At the initial frame ($m = 0$), the reference image embedding c^{ref} is encoded by a VAE encoder and a CLIP visual encoder (Radford et al., 2021). For subsequent frames ($m > 0$), we do not condition on a reference image (\emptyset) and inject Gaussian noise in its place.

The model is trained with the flow matching objective, extended to incorporate control signals:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, \mathbf{x}_t} [\|\mathbf{v}_\theta(\mathbf{c}, t, \mathbf{x}_t) - \mathbf{v}\|_2^2], \quad (7)$$

where \mathbf{c} denotes the full set of control inputs across the entire video, e.g., $\mathbf{c} = \{c_m\}_{m=0}^M$.

Distilling to real-time AR model. Following previous techniques (Huang et al., 2025; Yin et al., 2025), we distill the bidirectional teacher model into a few-step student model by replacing bidirectional attention with causal attention. The student is further optimized with DMD (Yin et al., 2024a) and adversarial losses (Goodfellow et al., 2020). Given a noise schedule $\mathcal{T} = \{t_0 = T, \dots, t_N = 0\}$, each frame is denoised over N steps, where N is significantly smaller than that in multi-step VDMs, enabling real-time inference.

Self-Rollout: Markovizing AR training. Although an AR VDM conditions on its own generated history at inference, AR training typically uses teacher forcing—each step conditions on ground-truth past frames rather than model outputs—creating a train–test mismatch (exposure bias) and breaking the Markov property required for RL. As illustrated in Fig. 2 (a), noise is added to the ground-truth frame, and the model predicts the corresponding vector field.

To address this issue, we propose a Self-Rollout strategy, which maintains a key–value (KV) memory cache storing previously denoised frames as causal context. As shown in Fig. 2 (b), frames are denoised sequentially from pure noise during training. Let $\mathbf{x}_{m,n}$ denote the m -th frame at denoising step n . For the m -th frame, we randomly sample a denoising step n , denoise step-by-step from $\mathbf{x}_{m,0}$ to $\mathbf{x}_{m,n}$, and compute the DMD loss in Eq. (5) and adversarial loss. We then continue denoising from $\mathbf{x}_{m,n}$ to $\mathbf{x}_{m,N}$ step-by-step, updating the KV cache with the generated clean frame $\mathbf{x}_{m,N}$. In this way, subsequent frames are conditioned on the self-generated KV cache rather than ground-truth history. In contrast, Self-Forcing (Huang et al., 2025) updates the KV cache by collapsing the denoising trajectory from $\mathbf{x}_{m,n}$ to $\mathbf{x}_{m,N}$ into a single step. Our step-by-step Rollout more faithfully matches inference dynamics and naturally integrates with RL–based training.

3.3 STEP 2: REINFORCEMENT LEARNING ON AR VDM

Our Self-Rollout strategy (Sec. 3.2) “Markovizes” AR training by conditioning on model-generated histories and the ODE-to-SDE conversion in Eq. (4) supplies the stochasticity. Taken together, these resolve the two obstacles to applying GRPO—it requires an MDP and stochastic rollouts. In the sequel, we first set notations and formulate the MDP underlying video generation.

Notations. Consider a video of $M+1$ frames, each denoised in N steps. We denote the m -th frame at denoising step n by $\mathbf{x}_{m,n}$. Let $\mathbf{x}_{<m,N} = \{\mathbf{x}_{0,N}, \dots, \mathbf{x}_{m-1,N}\}$ be the $m-1$ already denoised clean frames and $\mathbf{x}_{>m,0} = \{\mathbf{x}_{m+1,0}, \dots, \mathbf{x}_{M,0}\}$ the unprocessed, noise-initialized frames. At state (m, n) , the video snapshot is

$$\mathbf{X}_{m,n} = \underbrace{\mathbf{x}_{<m,N}}_{\text{fully generated}} \cup \underbrace{\{\mathbf{x}_{m,n}\}}_{\text{being denoised}} \cup \underbrace{\mathbf{x}_{>m,0}}_{\text{initial noise}}, \quad (8)$$

The final clean video is then $\mathbf{X}_{M,N} = \{\mathbf{x}_{0,N}, \dots, \mathbf{x}_{M,N}\}$. For autoregressive video generation, the denoising across frames produces a trajectory

$$\tau = \underbrace{\{\mathbf{X}_{0,0}, \mathbf{X}_{0,1}, \dots, \mathbf{X}_{0,N}\}}_{\text{trajectory of frame 0}} \underbrace{\{\mathbf{X}_{1,0}, \mathbf{X}_{1,1}, \dots, \mathbf{X}_{1,N}, \dots, \mathbf{X}_{M,0}, \mathbf{X}_{M,1}, \dots, \mathbf{X}_{M,N}\}}_{\text{trajectory of frame 1}} \underbrace{\dots}_{\text{trajectory of frame } M}. \quad (9)$$

Video generation as MDP. The denoising process in VDM can be formulated as a Markov decision process (MDP) (Liu et al., 2025; Xue et al., 2025):

- **State:** $\mathbf{s}_{m,n} \triangleq (\mathbf{c}_m, t_n, \mathbf{X}_{m,n})$, where \mathbf{c}_m is the control signals. The initial-state distribution is $p(\mathbf{s}_{0,0}) = p(\mathbf{c}, t_0, \mathbf{X}_{0,0}) = p(\mathbf{c}_0) \delta(t-t_0) \prod_{m=0}^M \mathcal{N}(\mathbf{x}_{m,0} \mid \mathbf{0}, \mathbf{I})$, i.e., the control \mathbf{c}_0 is drawn from its prior, t is fixed to t_0 , and all frames start from Gaussian. $\delta(\cdot)$ denotes the Dirac distribution.
- **Action:** $\mathbf{a}_{m,n} \triangleq \mathbf{x}_{m,n+1}$, i.e., the next denoised state of the m -th frame at step $n+1$. The policy is parameterized by the VDM with θ :

$$\mathbf{a}_{m,n} = \mathbf{x}_{m,n+1} \sim p_\theta(\cdot \mid \mathbf{c}_m, t_n, \mathbf{X}_{m,n}). \quad (10)$$

where stochasticity is introduced through the ODE-to-SDE conversion in Eq. (4).

- **Transition:** (1) *intra-frame transition*. Within a frame, the transition is deterministic given the current state and action: $p(\mathbf{s} \mid \mathbf{s}_{m,n}, \mathbf{a}_{m,n}) = \delta(\mathbf{s} - \mathbf{s}_{m,n+1})$. (2) *inter-frame transition*. When denoising of frame m is complete ($n = N$), the state transitions to the initial state of the next frame $m+1$:

$$\mathbf{s}_{m+1,0} = (\mathbf{c}_{m+1}, t_0, \mathbf{X}_{m+1,0}), \quad \text{where } \mathbf{X}_{m+1,0} = \mathbf{X}_{m,N} \text{ by definition.} \quad (11)$$

- **Reward function:** Rewards are provided only when a frame is fully denoised ($n = N$):

$$R(\mathbf{s}_{m,n}, \mathbf{a}_{m,n}) \triangleq R(\mathbf{x}_{m,N}, \mathbf{c}_m) = \mathbb{1}[n = N] \cdot (R_{\text{quality}}(\mathbf{x}_{m,N}) + R_{\text{motion}}(\mathbf{x}_{m,N}, \mathbf{c}_m)) \quad (12)$$

where $\mathbb{1}[\cdot]$ is the indicator function, R_{quality} measures perceptual fidelity and temporal smoothness and R_{motion} measures alignment with control signals. (We defer precise definitions to the sequel.)

GRPO for AR VDM. We extend GRPO framework to AR video generation. Under the MDP formulation, the AR VDM samples a group of G videos $\{\mathbf{X}_{M,N}^{(i)}\}_{i=1}^G$ along with their trajectories

270 $\{\tau^{(i)}\}_{i=1}^G$. The advantage of the i -th video is computed as:
 271

$$272 \hat{A}_{m,n}^{(i)} = \frac{R(\mathbf{x}_{m,N}^{(i)}, \mathbf{c}_m) - \text{mean}(\{R(\mathbf{x}_{m,N}^{(j)}, \mathbf{c}_m)\}_{j=1}^G)}{\text{std}(\{R(\mathbf{x}_{m,N}^{(j)}, \mathbf{c}_m)\}_{j=1}^G)}. \quad (13)$$

273
 274

275 The GRPO objective is defined as:
 276

$$277 \mathcal{L}_{\text{GRPO}}(\pi_\theta) = \mathbb{E}_{\mathbf{c}, \{\tau^{(i)}\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{c})} \left[\frac{1}{GMN} \sum_{i=1}^G \sum_{m=1}^M \sum_{n=1}^N \left(\min \left(r_{m,n}^{(i)}(\theta) \hat{A}_{m,n}^{(i)}, \text{clip}(r_{m,n}^{(i)}(\theta), 1 - \varepsilon, 1 + \varepsilon) \hat{A}_{m,n}^{(i)} \right) - \beta \text{KL}(\pi_\theta \| \pi_{\text{ref}}) \right) \right] \quad (14)$$

278
 279
 280

281 where the importance ratio is: $r_{m,n}^{(i)}(\theta) = p_\theta(\mathbf{x}_{m,n+1}^{(i)} | \mathbf{x}_{m,n}^{(i)}, \mathbf{c}_m) / p_{\theta_{\text{old}}}(\mathbf{x}_{m,n+1}^{(i)} | \mathbf{x}_{m,n}^{(i)}, \mathbf{c}_m)$.
 282

283 **Selective stochastic sampling.** GRPO requires stochasticity for advantage estimation and policy
 284 exploration, which we introduce via the ODE-to-SDE conversion. However, in video generation the
 285 Markov chain is extremely long, and applying SDE sampling at every denoising step induces very
 286 high variance in trajectory returns, which substantially increases the number of rollouts (G) needed
 287 for stable loss estimation and thus incurs prohibitive cost.
 288

289 To balance exploration and efficiency, we adopt *selective stochasticity*: a single denoising step \hat{n}
 290 is randomly chosen to follow the SDE formulation, while all remaining steps stay deterministic
 291 under the ODE solver. This strategy injects sufficient randomness for effective RL training, while
 292 maintaining computational efficiency.
 293

294 **Reward design.** We design a composite reward that jointly evaluates visual realism (R_{quality})
 295 and motion controllability (R_{motion}). For realism, we adopt the LAION Aesthetic Quality Predictor
 296 (Schuhmann, 2022) denoted as f_{AQ} that assigns an aesthetic score (1-5) to each image. The
 297 realism reward is defined as
 298

$$299 R_{\text{quality}}(\mathbf{x}_{m,N}) = f_{\text{AQ}}(\mathbf{x}_{m,N}). \quad (15)$$

300

301 For motion controllability, we employ Co-Tracker (Karaev et al., 2024) to first estimate the object
 302 trajectory $\hat{\mathbf{c}}_m^{\text{traj}}$ at frame m from the generated image and measure their alignment with the ground-
 303 truth $\mathbf{c}_m^{\text{traj}}$. The motion reward is defined as
 304

$$305 R_{\text{motion}}(\mathbf{x}_{m,N}, \mathbf{c}_m) = \lambda \max(0, \alpha - \|\hat{\mathbf{c}}_m^{\text{traj}} - \mathbf{c}_m^{\text{traj}}\|_2^2), \quad (16)$$

306 where α is an offset, and λ is the scaling hyperparameter.
 307

308 3.4 DISCUSSION WITH EXISTING TECHNIQUES.

309 Our Self-Rollout eliminates this collapse entirely by continuing full step-by-step ancestral sampling
 310 using only the model’s own predictions—identical to inference. Combined with selective stochasticity
 311 (Sec. 3.3), we reduce the effective horizon by 5–20× while preserving exploration, enabling
 312 stable and effective GRPO training on autoregressive video diffusion for the first time.
 313

314 **Comparison with Self-Forcing** Although our Self-Rollout strategy and Self-Forcing (Huang et al.,
 315 2025) both address exposure bias by using self-generated context in autoregressive video
 316 diffusion, they differ fundamentally in how the KV cache is updated after the supervised prefix. These
 317 differences critically impact alignment with inference-time dynamics and compatibility with rein-
 318 force learning objectives such as GRPO.
 319

320 By performing a full step-by-step rollout instead of a single non-sequential collapse, Self-Rollout
 321 perfectly eliminates the train–inference distribution mismatch, provides a clean sequential decision
 322 process that GRPO can directly optimize, and—when combined with selective stochasticity sam-
 323 pling—effectively mitigates the extremely long-horizon problem. This enables successful applica-
 324 tion of GRPO to high-fidelity autoregressive video generation for the first time.
 325

326 4 EXPERIMENTS

327 **Implementation details.** We implement our base model with Wan2.1-1.3B-I2V Wan et al. (2025),
 328 using a 3-step diffusion process in a frame-wise manner, denosing one latent at a time. To accommo-
 329 date varying resolutions, we define a set of bucket sizes and resize each video to its nearest bucket.
 330

324 Table 1: Quantitative comparisons with motion-controllable VDMs. Best results are **bold**.
325

Method	Latency (s) ↓	FID ↓	FVD ↓	Aesthetic Quality ↑	Motion Smoothness ↑	Motion Consistency ↑
DragNUWA	94.26	36.31	376.39	3.30	0.9759	3.71
DragAnything	68.76	38.13	367.74	3.22	0.9811	3.63
Tora	176.51	32.84	283.43	3.86	0.9855	3.97
MagicMotion	1426.37	30.04	230.53	4.01	0.9871	3.95
Self-Forcing	0.95	34.47	315.87	3.70	0.9920	4.06
AR-Drag	0.44	28.98	187.49	4.07	0.9948	4.37

334
335 The KV cache is set to hold 7 frames; when updating the cache, the oldest frame is removed if
336 the cache exceeds this size. All training is performed using the AdamW optimizer (Loshchilov &
337 Hutter, 2017) with a learning rate of 1×10^{-5} , on 8 NVIDIA H20 GPUs. For evaluation, we cu-
338 rate a new benchmark consisting of 206 video clips covering diverse motion trajectories and scene
339 variations, specifically designed to assess motion controllability.

340 **Metrics.** We adopt standard metrics such as Fréchet Inception Distance (FID) (Seitzer, 2020),
341 Fréchet Video Distance (FVD) (Unterthiner et al., 2018), and Aesthetic Quality (Schuhmann, 2022)
342 to quantitatively evaluate visual quality. To assess motion controllability, we employ two comple-
343 mentary measures: Motion Smoothness (Huang et al., 2024), which captures the stability of motion
344 across frames, and Motion Consistency, which evaluates the alignment between control trajectories
345 and the resulting motion dynamics, computed using our proposed reward model. We report first-
346 frame latency calculated on a single NVIDIA H20 GPU as an indicator of real-time performance.

347 **Baselines.** We compare our method against strong open-source motion-guided VDMs, including
348 DragNUWA (Yin et al., 2023), DragAnything (Wu et al., 2024), Tora (Zhang et al., 2025) and
349 Magicmotion (Li et al., 2025a). Following prior work (Zhang et al., 2025), we improve DragNUWA
350 by adopting its motion trajectory design to a DiT-based architecture. **Tora is the first one to apply**
351 **DiT in this task, and MagicMotion further support complex trajectories-based controls.** Since no AR
352 motion-control I2V baseline is available, we fine-tune a chunk-wise AR VDM, Self-Forcing (Huang
353 et al., 2025), which was originally designed for text-to-video (T2V) generation. Specifically, we
354 fine-tune Wan2.1-1.3B-I2V following the Self-Forcing architecture and training procedure using the
355 same datasets as AR-Drag. In this adaptation, the model denoises three latents simultaneously in
356 each denoising loop to achieve effective motion controllability.

357 4.1 RESULTS

358 **Quantitative comparisons.** The overall performance comparisons are reported in Tab. 1, leading to
359 the following key observations: Our method **significantly reduces latency**. It requires only 0.44s,
360 while bidirectional approaches such as Tora take 176.51s—less than 1% of their latency. **For the 5B**
361 **model MagicMotion, the latency is even higher at 1426.37 s.** Thanks to the few-step distillation and
362 causal design, our model can produce results immediately once the first frame is generated..

363 Despite being a few-step autoregressive design, AR-Drag still delivers **the best visual quality**.
364 Specifically, it achieves the lowest FID and FVD, as well as the highest Aesthetic Quality, re-
365 flecting superior visual fidelity and temporal coherence. In terms of motion control metrics, our
366 model attains the highest motion smoothness and consistency, highlighting its strength in precise
367 and stable motion control. This contributes to our RL post training, which incentivizes the model’s
368 ability to follow motion guidance, enabling more flexible and robust controllability. **Remarkably,**
369 **AR-Drag even outperforms the 5B MagicMotion, particularly on motion-control. MagicMotion**
370 **does not utilize RL training, which limits its ability to achieve fine-grained, highly flexible control.**

371 Self-Forcing baseline also adopts a few-step AR design, but requires 0.95s—more than twice our
372 latency—since it denoises three frames simultaneously. Moreover, AR-Drag outperforms Self-
373 Forcing in both visual quality and motion control. These results demonstrate the effectiveness of
374 our RL post-training and Self-Rollout for real-time motion-controllable video generation.

375 **Qualitative comparisons.** We conduct qualitative comparisons with three competitive baselines,
376 Tora, MagicMotion and Self-Forcing. As shown in Fig. 3, we evaluate across different prompts,



Figure 3: Qualitative comparisons with Tora and Self-Forcing across different prompts, data domains, and resolutions, demonstrating the superior fidelity and controllability of our method.

Table 2: Ablation study on key training strategies. ‘w/o RL’ denotes removing the RL post-training. ‘Initial model’ refers to Wan2.1-1.3B-I2V prior to adaptation. ‘Teacher model’ is the fine-tuned multi-step bidirectional model. ‘w/o Self-Rollout’ denotes training without the Self-Rollout design.

Method	Latency (s) \downarrow	FID \downarrow	FVD \downarrow	Aesthetic Quality \uparrow	Motion Smoothness \uparrow	Motion Consistency \uparrow
AR-Drag	0.44	28.98	187.49	4.07	0.9948	4.37
w/o RL	0.44	31.65	210.35	3.92	0.9926	4.12
Initial model	45.72	35.94	303.16	3.84	0.9915	3.22
Teacher model	45.64	29.38	151.46	4.15	0.9941	4.36
w/o Self-Rollout	0.44	38.13	353.75	3.38	0.9904	4.02

ranging from specific actions such as head shaking and taking off clothes, to more general motions such as following a trajectory. We further compare performance on both synthetic data (a), (c), (d) and real-world data (b), as well as across different resolutions. Since Tora only supports a fixed resolution, the resolution-based comparison in (c) and (d) is conducted only against Self-Forcing. For clarity, we visualize the entire trajectory across frames in blue and highlight the control signal of the current frame in red. The reference image is provided for the first frame. Since the same negative prompt is applied to all videos, only the positive prompt is shown.

As illustrated in Fig. 3(a&b), Tora and MagicMotion struggle to maintain consistency with the control signals. Self-Forcing achieves partial controllability but suffers from noticeable deformation and severe quality degradation. In contrast, our method delivers superior fidelity and control alignment. Furthermore, as shown in Fig. 3(c&d), Self-Forcing exhibits substantial detail loss—particularly in fine structures such as fingers and hair strands—and suffers from increased color saturation in (c), whereas our method consistently preserves high-quality details and maintains faithful motion control.

4.2 ABLATION STUDIES

In Tab. 2, we present the ablations on key training strategies.

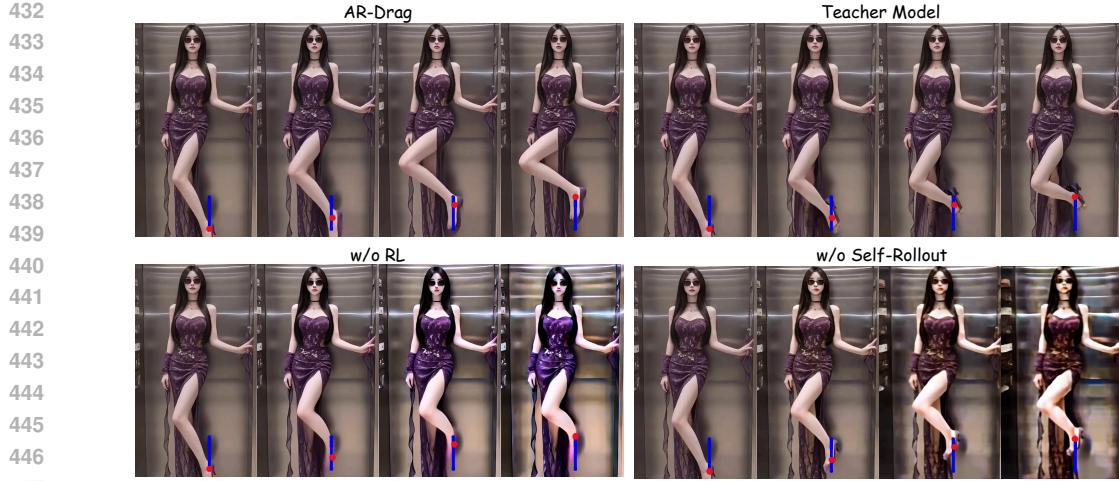


Figure 4: Ablation on key training strategies. Prompt: movement following the trajectory.

w/o RL. Removing reinforcement learning leads to a noticeable drop in both quality and motion-related metrics, highlighting the importance of RL in enhancing fidelity and motion controllability.

Initial model. The initial Wan2.1-1.3B-I2V model performs worse than our base model (w/o RL) on video quality and have a high latency, demonstrating that our motion fine-tuning and real-time post-training strategies provide a strong foundation for RL training.

Teacher model. The teacher model, a fine-tuned bidirectional multi-step baseline, achieves strong performance but suffers from high latency. While it represents the upper bound of DMD-based method, our AR-Drag achieves comparable or even better results in FID, Aesthetic Quality, Motion Smoothness, and Motion Consistency, confirming the effectiveness of our RL approach.

w/o Self-Rollout. Removing the Self-Rollout design leads to severe quality degradation, underscoring its necessity for maintaining the Markov property and mitigating the train-test mismatch in autoregressive generation.

Visualization. Since the initial model performs significantly worse, we exclude it from the comparison. As shown in Fig. 4, due to the absence of the feet in the reference image, both the teacher model and the model without RL fail to generate clear foot details, reflecting limited generalization. In contrast, our RL-based method encourages exploration, enhancing the model’s generalization capability. Additionally, the model w/o RL exhibits increased color saturation, while the model without Self-Rollout suffers from severe image artifacts and quality degradation, caused by the train–test discrepancy and the disruption of the Markov property.

Visualization on diverse motion. We show qualitative results of our model conditioned on different motion trajectories in Fig. 5. The results demonstrate that our method can accurately follow diverse motion commands, while preserving visual quality, and temporal consistency across frames.

5 DISCUSSION

To better clarify our contributions and limitations, we discuss the two key technical components of AR-Drag along with its primary limitation.

Difference between Self-Rollout and Self-Forcing. Applying GRPO to AR VDMs requires the base model to follow the Markov Decision Chain. However, standard AR VDMs exhibit a training–inference gap—training conditions on ground truth while inference conditions on generated frames—breaking the Markov property and preventing direct GRPO training. Self-Forcing partially reduces this gap by using self-generated frames as KV cache (Alg. 2), but it skips remaining denoising steps and thus still violates the MDP. Our Self-Rollout enforces the full denoising rollout for every frame, restoring the proper Markov structure. This simple but critical correction makes RL training valid and leads to clear performance gains, as shown in Table 1, Figure 3, and the ‘w/o Self-Rollout’ ablations in Table 2 and Figure 4.

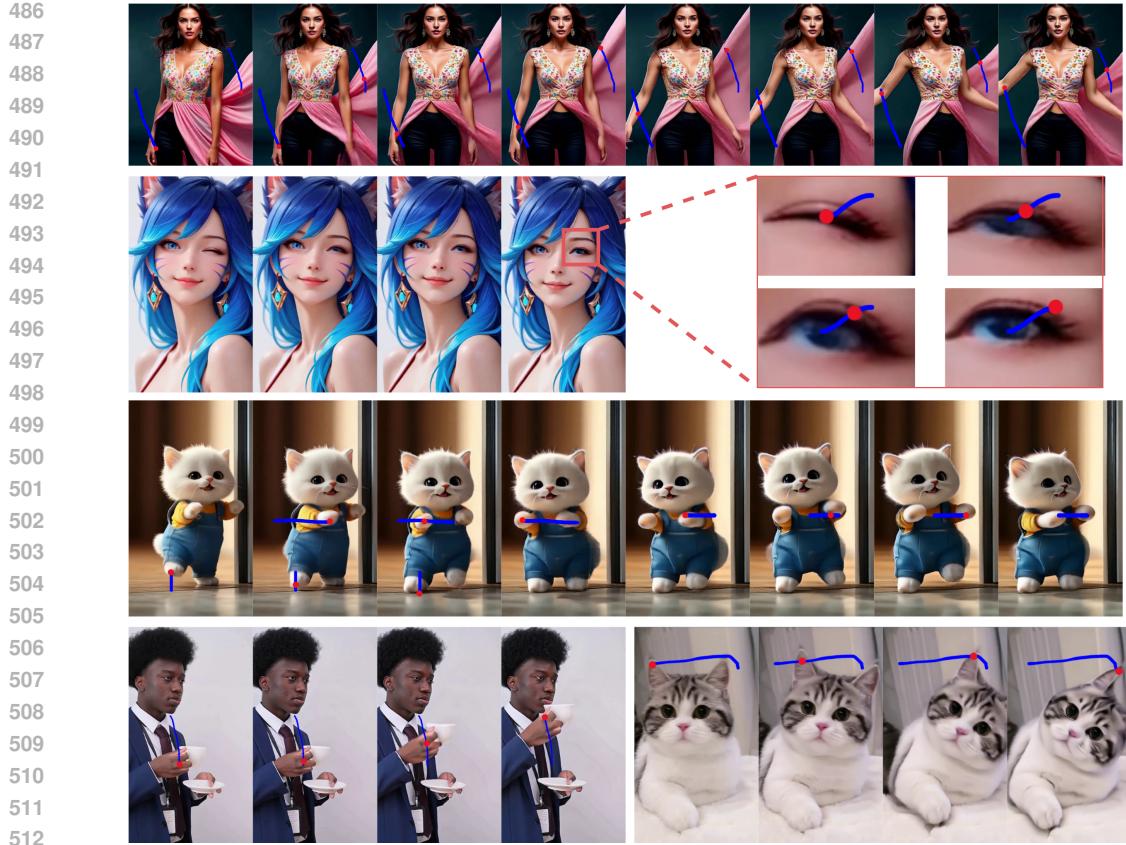


Figure 5: Visualization on diverse motion. Prompt: movement following the trajectory.

Importance of selective stochastic sampling. In bidirectional VDMs, frames are denoised jointly, so the decision-chain length equals the number of denoising steps. In AR VDMs, however, frames are denoised sequentially, making the chain length scale with denoising steps \times frame count, leading to extremely long horizons. This causes return variance to explode and gradient estimates to become unusably noisy, making direct GRPO (or any policy-gradient method) practically infeasible. Our selective stochasticity sampling provides controlled exploration at each step without triggering variance explosion, enabling stable and sample-efficient GRPO training for AR video generation.

Limitation of AR-Drag. Our generative model is trained on data that follows physical plausibility, and our reward model is also designed to evaluate motion based on physical principles. Therefore, if a user intentionally provides highly exaggerated or physically impossible control signals, the model may not strictly follow such inputs because they fall outside the distribution it is trained and rewarded to respect. Handling deliberately non-physical or cartoon-like motion is an interesting direction for future work, and we believe extending controllability beyond physically plausible dynamics is a valuable avenue for exploration.

6 CONCLUSION

We present AR-Drag, the first RL-enhanced few-step autoregressive video diffusion model for real-time motion-controllable image-to-video generation. By combining selective stochasticity, and a trajectory-based reward model, our approach effectively addresses the challenges of quality degradation, motion artifacts, and complex control spaces in few-step AR video generation. Extensive experiments demonstrate that AR-Drag achieves high visual fidelity, precise motion alignment, and significantly lower latency compared with state-of-the-art motion-controllable VDMs, while maintaining a compact model size of only 1.3B parameters.

540 ETHICS STATEMENT
541542 This work presents a method for real-time controllable video generation. Our experiments are con-
543 ducted on de-identified datasets that do not contain personally identifiable information. The study is
544 intended solely for scientific research, and we adhere to the ICLR Code of Ethics regarding fairness,
545 integrity, and responsible use of data and models.546
547 REPRODUCIBILITY STATEMENT
548549 We provide detailed implementation settings, including model architecture, training objectives, op-
550 timization strategies, and hyperparameters in the main text and Appendix. The code, configuration
551 files, and instructions for reproducing the main experiments are available in Supplementary Materi-
552 als to facilitate verification and further research.553
554 REFERENCES
555556 Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik
557 Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling
558 latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023a.559 Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler,
560 and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion
561 models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*,
562 pp. 22563–22575, 2023b.563 Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe
564 Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI*
565 *Blog*, 1(8):1, 2024.566 Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitz-
567 mann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in*
568 *Neural Information Processing Systems*, 37:24081–24125, 2024.569 Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models
570 on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.571 Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao
572 Carreira, Andrew Zisserman, and Yi Yang. TAP-vid: A benchmark for tracking any point in a
573 video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022.574 Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao,
575 Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative
576 foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.577 Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel,
578 Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-
579 tuning text-to-image diffusion models. In *Thirty-seventh Conference on Neural Information Pro-
580 cessing Systems (NeurIPS) 2023*. Neural Information Processing Systems Foundation, 2023.581 Hiroki Furuta, Heiga Zen, Dale Schuurmans, Aleksandra Faust, Yutaka Matsuo, Percy Liang, and
582 Sherry Yang. Improving dynamic object interactions in text-to-video generation with ai feedback.
583 *arXiv preprint arXiv:2412.02617*, 2024.584 Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-vdm:
585 Efficient autoregressive video diffusion model with causal generation and cache sharing. *arXiv*
586 *preprint arXiv:2411.16375*, 2024.587 Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana
588 Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, et al. Motion prompting: Control-
589 ling video generation with motion trajectories. In *Proceedings of the Computer Vision and Pattern*
590 *Recognition Conference*, pp. 1–12, 2025.

594 Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,
 595 Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the*
 596 *ACM*, 63(11):139–144, 2020.

597

598 Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with
 599 next-frame prediction. *arXiv preprint arXiv:2503.19325*, 2025.

600 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
 601 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
 602 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

603

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

607 Jinyi Hu, Shengding Hu, Yuxuan Song, Yufei Huang, Mingxuan Wang, Hao Zhou, Zhiyuan Liu,
 608 Wei-Ying Ma, and Maosong Sun. Acdit: Interpolating autoregressive conditional modeling and
 609 diffusion transformer. *arXiv preprint arXiv:2412.07720*, 2024.

610

611 Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging
 612 the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

613 Ziqi Huang, Yinan He, Jiahuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianx-
 614 ing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for
 615 video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*
 616 *Pattern Recognition*, pp. 21807–21818, 2024.

617

618 Hyeonho Jeong, Geon Yeong Park, and Jong Chul Ye. Vmc: Video motion customization using
 619 temporal attention adaption for text-to-video diffusion models. In *Proceedings of the IEEE/CVF*
 620 *Conference on Computer Vision and Pattern Recognition*, pp. 9212–9221, 2024.

621 Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song,
 622 Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling.
 623 *arXiv preprint arXiv:2410.05954*, 2024.

624

625 Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian
 626 Rupprecht. Cotracker: It is better to track together. In *European conference on computer vision*,
 627 pp. 18–35. Springer, 2024.

628

629 Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li,
 630 Bo Wu, Jianwei Zhang, et al. Hunyuandvideo: A systematic framework for large video generative
 631 models. *arXiv preprint arXiv:2412.03603*, 2024.

632

633 Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel,
 634 Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human
 635 feedback. *arXiv preprint arXiv:2302.12192*, 2023.

636

637 Quanhao Li, Zhen Xing, Rui Wang, Hui Zhang, Qi Dai, and Zuxuan Wu. Magicmotion: Controllable
 638 video generation with dense-to-sparse trajectory guidance. *arXiv preprint arXiv:2503.16421*,
 2025a.

639

640 Zongyi Li, Shujie Hu, Shujie Liu, Long Zhou, Jeongsoo Choi, Lingwei Meng, Xun Guo, Jinyu Li,
 641 Hefei Ling, and Furu Wei. Arlon: Boosting diffusion transformers with autoregressive models
 642 for long video generation. In *ICLR*, 2025b.

643

644 Shanchuan Lin, Ceyuan Yang, Hao He, Jianwen Jiang, Yuxi Ren, Xin Xia, Yang Zhao, Xuefeng
 645 Xiao, and Lu Jiang. Autoregressive adversarial post-training for real-time interactive video gen-
 646 eration. *arXiv preprint arXiv:2506.09350*, 2025.

647

648 Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan,
 649 Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv*
 650 *preprint arXiv:2505.05470*, 2025.

648 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint*
 649 *arXiv:1711.05101*, 2017.

650

651 Wan-Duo Kurt Ma, John P Lewis, and W Bastiaan Kleijn. Trailblazer: Trajectory control for
 652 diffusion-based video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

653

654 Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. Revideo:
 655 Remake a video with motion and content control. *Advances in Neural Information Processing*
 656 *Systems*, 37:18481–18505, 2024.

657

658 Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression:
 659 Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

660

661 Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-
 662 image diffusion models with reward backpropagation. 2023.

663

664 Mihir Prabhudesai, Russell Mendonca, Zheyang Qin, Katerina Fragkiadaki, and Deepak Pathak.
 665 Video diffusion alignment via reward gradients. *arXiv preprint arXiv:2407.08737*, 2024.

666

667 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,
 668 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual
 669 models from natural language supervision. In *International conference on machine learning*, pp.
 670 8748–8763. PMLR, 2021.

671

672 Rafael Rafailev, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea
 673 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances*
 674 *in neural information processing systems*, 36:53728–53741, 2023.

675

676 Christoph Schuhmann. Laion aesthetics, Aug 2022.

677

678 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy
 679 optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

680

681 Maximilian Seitzer. pytorch-fid: Fid score for pytorch. <https://github.com/mseitzer/pytorch-fid>, 2020.

682

683 Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang,
 684 Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable
 685 image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference*
 686 *Papers*, pp. 1–11, 2024.

687

688 Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv*
 689 *preprint arXiv:2310.14189*, 2023.

690

691 Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

692

693 Mingzhen Sun, Weining Wang, Gen Li, Jiawei Liu, Jiahui Sun, Wanquan Feng, Shanshan Lao, SiYu
 694 Zhou, Qian He, and Jing Liu. Ar-diffusion: Asynchronous video generation with auto-regressive
 695 diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7364–
 696 7373, 2025.

697

698 Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning
 699 Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv*
 700 *preprint arXiv:2505.13211*, 2025.

701

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

705

706 Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang,
 707 Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable
 708 length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*,
 709 2022.

702 Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam,
 703 Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using
 704 direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision
 705 and Pattern Recognition*, pp. 8228–8238, 2024.

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

709 Hanlin Wang, Hao Ouyang, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Qifeng Chen, Yujun Shen,
 710 and Limin Wang. Levitor: 3d trajectory oriented image-to-video synthesis. In *Proceedings of the
 711 Computer Vision and Pattern Recognition Conference*, pp. 12490–12500, 2025.

712 Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen,
 713 Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion con-
 714 trollability. *Advances in Neural Information Processing Systems*, 36:7594–7611, 2023.

715 Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo,
 716 and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In
 717 *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

718 Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou,
 719 Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity
 720 representation. In *European Conference on Computer Vision*, pp. 331–348. Springer, 2024.

721 Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao
 722 Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation.
 723 *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

724 Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei
 725 Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv
 726 preprint arXiv:2505.07818*, 2025.

727 Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li,
 728 Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation.
 729 *arXiv preprint arXiv:2509.22622*, 2025.

730 Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang,
 731 Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models
 732 with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

733 Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Drag-
 734 nuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv
 735 preprint arXiv:2308.08089*, 2023.

736 Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and
 737 Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in
 738 neural information processing systems*, 37:47455–47487, 2024a.

739 Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman,
 740 and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of
 741 the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024b.

742 Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and
 743 Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceed-
 744 ings of the Computer Vision and Pattern Recognition Conference*, pp. 22963–22974, 2025.

745 Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and
 746 Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. In *Proceed-
 747 ings of the Computer Vision and Pattern Recognition Conference*, pp. 2063–2073, 2025.

748 Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jia-Wei Liu, Weijia Wu, Jussi Keppo,
 749 and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models.
 750 In *European Conference on Computer Vision*, pp. 273–290. Springer, 2024.

756 **Algorithm 1** Self-Rollout Training

757 **Require:** Denoising schedule $\{t_0, t_1, \dots, t_N\}$, Number of frames $M + 1$, model G_θ ,
 758 $\{\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_N\}$

759 1: **loop**

760 2: Initialize model output $\mathbf{X}_\theta \leftarrow []$, $KVcache KV \leftarrow []$

761 3: Sample $s \sim \text{Unif}\{1, \dots, N\}$

762 4: **for** $m = 0$ **to** M **do**

763 5: Initialize $\mathbf{x}_{m,0} \sim \mathcal{N}(\mathbf{0}, I)$

764 6: **for** $n = 0$ **to** s **do**

765 7: **if** $n = s$ **then** ▷ Ensure all denoising steps could be optimized

766 8: Enable gradient computation

767 9: $\hat{\mathbf{x}}_{m,N} \leftarrow G_\theta(\mathbf{x}_{m,s}, t_s, \mathbf{c}_m, KV)$

768 10: $\mathbf{X}_\theta.append(\hat{\mathbf{x}}_{m,N})$

769 11: **else**

770 12: Disable gradient computation

771 13: $\hat{\mathbf{x}}_{m,N} \leftarrow G_\theta(\mathbf{x}_{m,n}, t_n, \mathbf{c}_m, KV)$

772 14: Sample $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$

773 15: $\hat{\mathbf{x}}_{m,k-1} \leftarrow \Psi(\hat{\mathbf{x}}_{m,N}, \epsilon, t_{k-1})$

774 16: **end if**

775 17: **end for**

776 18: $\hat{\mathbf{x}}_{m,s+1} \leftarrow \Psi(\hat{\mathbf{x}}_{m,N}, \epsilon, t_{s+1})$

777 19: **for** $n = s + 1$ **to** N **do**

778 20: **if** $n = s$ **then** ▷ Enforce the MDP defined in Eq. 9

779 21: $\hat{\mathbf{x}}_{m,N} \leftarrow G_\theta(\mathbf{x}_{m,s}, t_s, \mathbf{c}_m, KV)$

780 22: $kv_m \leftarrow G_\theta(\hat{\mathbf{x}}_{m,N}, KV)$ ▷ Update KV cache with the right generation

781 23: $KV.append(kv_m)$

782 24: **else**

783 25: $\hat{\mathbf{x}}_{m,N} \leftarrow G_\theta(\mathbf{x}_{m,n}, t_n, \mathbf{c}_m, KV)$

784 26: Sample $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$

785 27: $\hat{\mathbf{x}}_{m,n+1} \leftarrow \Psi(\hat{\mathbf{x}}_{m,N}, \epsilon, t_{n+1})$

786 28: **end if**

787 29: **end for**

788 30: **end for**

789 31: Update θ on \mathbf{X}_θ

790 32: **end loop**

791 **A MORE DETAILS ABOUT ARCHITECTURE**

792 To better illustrates the difference between Self-Rollout and Self-Forcing, we provide the detailed
 793 training process in Alg. 1 and Alg. 2, respectively. As shown in Line 7 of Alg. 2, Self-Forcing
 794 randomly selects t_s from denoising schedule and use the output at this step to compute loss (Line
 795 9), ensuring that all denoising steps could be optimized. However, it directly treats the output at
 796 this intermediate denoising step as the final generated clean frame (Line 8), and uses it to update the
 797 KV cache (Line 10). This skips the remaining denoising steps from s to N , which breaks the MDP
 798 defined in Eq. 9. And the incorrect KV cache subsequently affects future generation. To address
 799 this issue, our Self-Rollout continues the denoising process all the way to step N , enforcing the full
 800 MDP transition defined in Eq. 9. This complete rollout is implemented in Lines 18–28 of Alg. 1.

801 In addition, we also provide the pseudo-code of Self-Rollout and Self-Forcing in Listing 1 and 2.

804 **B MORE EXPERIMENTAL SETTINGS**805 **B.1 DATA CURATION**

806 We construct our training corpus by combining both real and synthetic videos to cover diverse motion
 807 patterns. For the real videos, we directly collect footage from real-world sources. For the synthetic
 808 videos, we use Wan2.1-14B-I2V to generate videos containing a wide range of motion types

810 **Algorithm 2** Self-Forcing Training

```

811 Require: Denoising schedule  $\{t_0, t_1, \dots, t_N\}$ , Number of frames  $M+1$ , model  $G_\theta$ , control signals
812    $\{\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_N\}$ 
813   1: loop
814     2: Initialize model output  $\mathbf{X}_\theta \leftarrow []$ ,  $KVcache KV \leftarrow []$ 
815     3: Sample  $s \sim \text{Unif}\{1, \dots, N\}$ 
816     4: for  $m = 0$  to  $M$  do
817       5: Initialize  $\mathbf{x}_{m,0} \sim \mathcal{N}(\mathbf{0}, I)$ 
818       6: for  $n = 0$  to  $s$  do
819         7: if  $n = s$  then            $\triangleright$  One-step collapse from s to N, which skip steps in MDP
820           8:  $\hat{\mathbf{x}}_{m,N} \leftarrow G_\theta(\mathbf{x}_{m,s}, t_s, \mathbf{c}_m, KV)$ 
821           9:  $\mathbf{X}_\theta.append(\hat{\mathbf{x}}_{m,N})$ 
822          10:  $\mathbf{kv}_m \leftarrow G_\theta(\hat{\mathbf{x}}_{m,N}, KV)$        $\triangleright$  Update KV cache with the collapsed generation
823          11:  $KV.append(\mathbf{kv}_m)$ 
824         else
825           13:  $\hat{\mathbf{x}}_{m,N} \leftarrow G_\theta(\mathbf{x}_{m,n}, t_n, \mathbf{c}_m, KV)$ 
826           14: Sample  $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$ 
827           15:  $\hat{\mathbf{x}}_{m,n+1} \leftarrow \Psi(\hat{\mathbf{x}}_{m,N}, \epsilon, t_{n+1})$ 
828         end if
829       17: end for
830     18: end for
831     19: Update  $\theta$  on  $\mathbf{X}_\theta$ 
832   20: end loop
833

```

834 but without control signals. In total, we gather approximately 10,000 videos. We then generate
835 trajectory-based control signals using an automatic detector, followed by manual filtering to remove
836 videos containing sensitive content, low-quality samples, or incorrect trajectories. For challenging
837 scenarios, such as severe occlusion or fast motion, we curate a high-quality subset of approximately
838 3,000 videos, all of which are fully annotated by human annotators. Control signals include motion
839 trajectories, prompts, and reference images. For motion trajectories, to better simulate actual user
840 interactions, we represent each point as a bright spot with intensity ranging from 0 to 1 rather than a
841 single isolated coordinate, mimicking the user’s touch force on each frame.

842 For prompts, we provide both negative and positive prompts. The negative prompt is shared across
843 all videos and follows the template:

844 **Negative Prompt Template**

845 *Overly vivid colors, overexposed, static, blurry details, subtitles, style, artwork, frame, still,
846 overall grayish, worst quality, low quality, JPEG compression artifacts, ugly, incomplete,
847 extra fingers, poorly drawn hands, poorly drawn faces, deformed, disfigured, malformed
848 limbs, fused fingers, motionless frame, cluttered background, three legs, many people in the
849 background, walking upside down.*

850
851 For positive prompts, we include either general motions along trajectories or specific actions to guide
852 the desired video content.

853 To handle videos of varying resolutions, we define a set of predefined “bucket sizes” and resize
854 each input video to its nearest bucket. The buckets include resolutions such as 480×368, 400×400,
855 368×480, 640×368, and 368×640. This strategy ensures consistent input dimensions while preserv-
856 ing aspect ratios as much as possible.

857
858 **B.2 IMPLEMENTATION DETAILS**

859 We implement our base model using Wan2.1-1.3B-I2V Wan et al. (2025), employing a 3-step dif-
860 fusion process with $N = 3$, and timesteps $t_0 = 1000, t_1 = 755, t_2 = 522, t_3 = 0$. We set chunk
861 size as 1, cache size as 7. For distillation post training, we set DMD loss weight as 1, generator loss
862 weight as 0.1, discriminator loss as 0.05.

```

864
865     1 def self_rollout_training(model, x_gt, cond_list, schedule):
866     2     # schedule = [t_0, t_1, ..., t_N], len(schedule) = N+1
867     3     # cond_list = [c_0, c_1, ..., c_M]
868     4     M = len(cond_list) - 1
869     5     X_theta = []           # collect supervised clean predictions
870     6     KV = []              # KV cache
871
872
873     7     # Sample random supervised prefix length
874     8     s = random.randint(0, N)           # Unif{0, ..., N}
875
876
877     9     for m in range(M+1):
878    10         c_m = cond_list[m]
879    11         x = torch.randn_like(x_gt[m])    # x_{m,0} ~ N(0, I)
880
881
882    12         # ===Phase 1: Supervised prefix (0 to s) ===
883    13         for n in range(s + 1):          # n = 0,1,...,s
884    14             if n == s:
885
886    15                 # Last supervised step: gradient flows
887    16                 torch.enable_grad()
888    17                 # predict clean frame
889    18                 x_hat_N = model(x, schedule[n], c_m, KV)
890    19                 X_theta.append(x_hat_N)
891
892    20             else:
893
894    21                 torch.no_grad()
895    22                 x_hat_N = model(x, schedule[n], c_m, KV)
896    23                 epsilon = torch.randn_like(x_hat_N)
897    24                 x = reverse_step(x_hat_N, epsilon, schedule[n+1])
898    25                 x = reverse_step(x_hat_N, epsilon, schedule[s+1])
899
900    26         # ===Phase 2: Self-generated rollout (s+1 to N) ===
901
902    27         torch.no_grad()
903    28         for n in range(s + 1, N + 1):
904
905    29             if n == N:
906
907    30                 x_hat_N = model(x, schedule[n], c_m, KV)
908    31                 # extract KV from clean frame
909    32                 kv_m = model.get_kv(x_hat_N, KV)
910    33                 KV.append(kv_m)
911
912    34             else:
913
914    35                 x_hat_N = model(x, schedule[n], c_m, KV)
915    36                 epsilon = torch.randn_like(x_hat_N)
916    37                 x = reverse_step(x_hat_N, epsilon, schedule[n+1])
917
918
919    38         # Update model using DMD loss on collected clean frames
920    39         loss = loss_func(X_theta, x_gt)
921    40         loss.backward()
922    41         optimizer.step()
923    42         optimizer.zero_grad()
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2145
2146
2147
2147
2148
2149
2149
2150
2151
2152
2152
2153
2153
2154
2154
2155
2155
2156
2156
2157
2157
2158
2158
2159
2159
2160
2160
2161
2161
2162
2162
2163
2163
2164
2164
2165
2165
2166
2166
2167
2167
2168
2168
2169
2169
2170
2170
2171
2171
2172
2172
2173
2173
2174
2174
2175
2175
2176
2176
2177
2177
2178
2178
2179
2179
2180
2180
2181
2181
2182
2182
2183
2183
2184
2184
2185
2185
2186
2186
2187
2187
2188
2188
2189
2189
2190
2190
2191
2191
2192
2192
2193
2193
2194
2194
2195
2195
2196
2196
2197
2197
2198
2198
2199
2199
2200
2200
2201
2201
2202
2202
2203
2203
2204
2204
2205
2205
2206
2206
2207
2207
2208
2208
2209
2209
2210
2210
2211
2211
2212
2212
2213
2213
2214
2214
2215
2215
2216
2216
2217
2217
2218
2218
2219
2219
2220
2220
2221
2221
2222
2222
2223
2223
2224
2224
2225
2225
2226
2226
2227
2227
2228
2228
2229
2229
2230
2230
2231
2231
2232
2232
2233
2233
2234
2234
2235
2235
2236
2236
2237
2237
2238
2238
2239
2239
2240
2240
2241
2241
2242
2242
2243
2243
2244
2244
2245
2245
2246
2246
2247
2247
2248
2248
2249
2249
2250
2250
2251
2251
2252
2252
2253
2253
2254
2254
2255
2255
2256
2256
2257
2257
2258
2258
2259
2259
2260
2260
2261
2261
2262
2262
2263
2263
2264
2264
2265
2265
2266
2266
2267
2267
2268
2268
2269
2269
2270
2270
2271
2271
2272
2272
2273
2273
2274
2274
2275
2275
2276
2276
2277
2277
2278
2278
2279
2279
2280
2280
2281
2281
2282
2282
2283
2283
2284
2284
2285
2285
2286
2286
2287
2287
2288
2288
2289
2289
2290
2290
2291
2291
2292
2292
2293
2293
2294
2294
2295
2295
2296
2296
2297
2297
2298
2298
2299
2299
2300
2300
2301
2301
2302
2302
2303
2303
2304
2304
2305
2305
2306
2306
2307
2307
2308
2308
2309
2309
2310
2310
2311
2311
2312
2312
2313
2313
2314
2314
2315
2315
2316
2316
2317
2317
2318
2318
2319
2319
2320
2320
2321
2321
2322
2322
2323
2323
2324
2324
2325
2325
2326
2326
2327
2327
2328
2328
2329
2329
2330
2330
2331
2331
2332
2332
2333
2333
2334
2334
2335
2335
2336
2336
2337
2337
2338
2338
2339
2339
2340
2340
2341
2341
2342
2342
2343
2343
2344
2344
2345
2345
2346
2346
2347
2347
2348
2348
2349
2349
2350
2350
2351
2351
2352
2352
2353
2353
2354
2354
2355
2355
2356
2356
2357
2357
2358
2358
2359
2359
2360
2360
2361
2361
2362
2362
2363
2363
2364
2364
2365
2365
2366
2366
2367
2367
2368
2368
2369
2369
2370
2370
2371
2371
2372
2372
2373
2373
2374
2374
2375
2375
2376
2376
2377
2377
2378
2378
2379
2379
2380
2380
2381
2381
2382
2382
2383
2383
2384
2384
2385
2385
2386
2386
2387
2387
2388
2388
2389
2389
2390
2390
2391
2391
2392
2392
2393
2393
2394
2394
2395
2395
2396
2396
2397
2397
2398
2398
2399
2399
2400
2400
2401
2401
2402
2402
2403
2403
2404
2404
2405
2405
2406
2406
2407
2407
2408
2408
2409
2409
2410
2410
2411
2411
2412
2412
2413
2413
2414
2414
2415
2415
2416
2416
2417
2417
2418
2418
2419
2419
2420
2420
2421
2421
2422
2422
2423
2423
2424
2424
2425
2425
2426
2426
2427
2427
2428
2428
2429
2429
2430
2430
2431
2431
2432
2432
2433
2433
2434
2434
2435
2435
2436
2436
2437
2437
2438
2438
2439
2439
2440
2440
2441
2441
2442
2442
2443
2443
2444
2444
2445
2445
2446
2446
2447
2447
2448
2448
2449
2449
2450
2450
2451
2451
2452
2452
2453
2453
2454
2454
2455
2455
2456
2456
2457
2457
2458
2458
2459
2459
2460
2460
2461
2461
2462
2462
2463
2463
2464
2464
2465
2465
2466
2466
2467
2467
2468
2468
2469
2469
2470
2470
2471
2471
2472
2472
2473
2473
2474
2474
2475
2475
2476
2476
2477
2477
2478
2478
2479
2479
2480
2480
2481
2481
2482
2482
2483
2483
2484
2484
2485
2485
2486
2486
2487
2487
2488
2488
2489
2489
2490
2490
2491
2491
2492
2492
2493
2493
2494
2494
2495
2495
2496
2496
2497
2497
2498
2498
2499
2499
2500
2500
2501
2501
2502
2502
2503
2503
2504
2504
2505
2505
2506
2506
2507
2507
2508
2508
2509
2509
2510
2510
2511
2511
2512
2512
2513
2513
2514
2514
2515
2515
2516
2516
2517
2517
2518
2518
2519
2519
2520
2520
2521
2521
2522
2522
2523
2523
2524
2524
2525
2525
2526
2526
2527
2527
2528
2528
2529
2529
2530
2530
2531
2531
2532
2532
2533
2533
2534
2534
2535
2535
2536
2536
2537
2537
2538
2538
2539
2539
2540
2540
2541
2541
2542
2542
2543
2543
2544
2544
2545
2545
2546
2546
2547
2547
2548
2548
2549
2549
2550
2550
2551
2551
2552
2552
2553
2553
2554
2554
2555
2555
2556
2556
2557
2557
2558
2558
2559
2559
2560
2560
2561
2561
2562
2562
2563
2563
2564
2564
2565
2565
2566
2566
2567
2567
2568
2568
2569
2569
2570
2570
2571
2571
2572
2572
2573
2573
2574
2574
2575
2575
2576
2576
2577
2577
2578
2578
2579
2579
2580
2580
2581
2581
2582
2582
2583
2583
2584
2584
2585
2585
2586
2586
2587
2587
2588
2588

```

```

918
919 1 def self_rollout_training(model, x_gt, cond_list, schedule):
920 2     # schedule = [t_0, t_1, ..., t_N], len(schedule) = N+1
921 3     # cond_list = [c_0, c_1, ..., c_M]
922 4     M = len(cond_list) - 1
923 5     X_theta = []           # collect supervised clean predictions
924 6     KV = []                # KV cache
925 7
926 8     # Sample random supervised prefix length
927 9     s = random.randint(0, N)           # Unif{0, ..., N}
928 10
929 11     for m in range(M+1):
930 12         c_m = cond_list[m]
931 13         x = torch.randn_like(x_gt[m])    # x_{m,0} ~ N(0, I)
932 14
933 15         # ===Phase 1: Supervised prefix (0 to s) ===
934 16         for n in range(s + 1):           # n = 0,1,...,s
935 17             if n == s:
936 18                 # Last supervised step: gradient flows
937 19                 torch.enable_grad()
938 20                 # predict clean frame
939 21                 x_hat_N = model(x, schedule[n], c_m, KV)
940 22                 X_theta.append(x_hat_N)
941 23                 # extract KV from collapsed generation
942 24                 kv_m = model.get_kv(x_hat_N, KV)
943 25                 KV.append(kv_m)
944 26
945 27             else:
946 28                 torch.no_grad()
947 29                 x_hat_N = model(x, schedule[n], c_m, KV)
948 30                 epsilon = torch.randn_like(x_hat_N)
949 31                 x = reverse_step(x_hat_N, epsilon, schedule[n+1])
950 32                 x = reverse_step(x_hat_N, epsilon, schedule[s+1])
951 33
952 34
953 35     # Update model using DMD loss on collected clean frames
954 36     loss = loss_func(X_theta, x_gt)
955 37     loss.backward()
956 38     optimizer.step()
957 39     optimizer.zero_grad()
958 40
959 41
960 42
961 43
962 44
963 45
964 46
965 47
966 48
967 49
968 50
969 51
970 52
971 53
972 54
973 55
974 56
975 57
976 58
977 59
978 60
979 61
980 62
981 63
982 64
983 65
984 66
985 67
986 68
987 69
988 70
989 71
990 72
991 73
992 74
993 75
994 76
995 77
996 78
997 79
998 80
999 81
999 82
999 83
999 84
999 85
999 86
999 87
999 88
999 89
999 90
999 91
999 92
999 93
999 94
999 95
999 96
999 97
999 98
999 99
999 100
999 101
999 102
999 103
999 104
999 105
999 106
999 107
999 108
999 109
999 110
999 111
999 112
999 113
999 114
999 115
999 116
999 117
999 118
999 119
999 120
999 121
999 122
999 123
999 124
999 125
999 126
999 127
999 128
999 129
999 130
999 131
999 132
999 133
999 134
999 135
999 136
999 137
999 138
999 139
999 140
999 141
999 142
999 143
999 144
999 145
999 146
999 147
999 148
999 149
999 150
999 151
999 152
999 153
999 154
999 155
999 156
999 157
999 158
999 159
999 160
999 161
999 162
999 163
999 164
999 165
999 166
999 167
999 168
999 169
999 170
999 171
999 172
999 173
999 174
999 175
999 176
999 177
999 178
999 179
999 180
999 181
999 182
999 183
999 184
999 185
999 186
999 187
999 188
999 189
999 190
999 191
999 192
999 193
999 194
999 195
999 196
999 197
999 198
999 199
999 200
999 201
999 202
999 203
999 204
999 205
999 206
999 207
999 208
999 209
999 210
999 211
999 212
999 213
999 214
999 215
999 216
999 217
999 218
999 219
999 220
999 221
999 222
999 223
999 224
999 225
999 226
999 227
999 228
999 229
999 230
999 231
999 232
999 233
999 234
999 235
999 236
999 237
999 238
999 239
999 240
999 241
999 242
999 243
999 244
999 245
999 246
999 247
999 248
999 249
999 250
999 251
999 252
999 253
999 254
999 255
999 256
999 257
999 258
999 259
999 260
999 261
999 262
999 263
999 264
999 265
999 266
999 267
999 268
999 269
999 270
999 271
999 272
999 273
999 274
999 275
999 276
999 277
999 278
999 279
999 280
999 281
999 282
999 283
999 284
999 285
999 286
999 287
999 288
999 289
999 290
999 291
999 292
999 293
999 294
999 295
999 296
999 297
999 298
999 299
999 300
999 301
999 302
999 303
999 304
999 305
999 306
999 307
999 308
999 309
999 310
999 311
999 312
999 313
999 314
999 315
999 316
999 317
999 318
999 319
999 320
999 321
999 322
999 323
999 324
999 325
999 326
999 327
999 328
999 329
999 330
999 331
999 332
999 333
999 334
999 335
999 336
999 337
999 338
999 339
999 340
999 341
999 342
999 343
999 344
999 345
999 346
999 347
999 348
999 349
999 350
999 351
999 352
999 353
999 354
999 355
999 356
999 357
999 358
999 359
999 360
999 361
999 362
999 363
999 364
999 365
999 366
999 367
999 368
999 369
999 370
999 371
999 372
999 373
999 374
999 375
999 376
999 377
999 378
999 379
999 380
999 381
999 382
999 383
999 384
999 385
999 386
999 387
999 388
999 389
999 390
999 391
999 392
999 393
999 394
999 395
999 396
999 397
999 398
999 399
999 400
999 401
999 402
999 403
999 404
999 405
999 406
999 407
999 408
999 409
999 410
999 411
999 412
999 413
999 414
999 415
999 416
999 417
999 418
999 419
999 420
999 421
999 422
999 423
999 424
999 425
999 426
999 427
999 428
999 429
999 430
999 431
999 432
999 433
999 434
999 435
999 436
999 437
999 438
999 439
999 440
999 441
999 442
999 443
999 444
999 445
999 446
999 447
999 448
999 449
999 450
999 451
999 452
999 453
999 454
999 455
999 456
999 457
999 458
999 459
999 460
999 461
999 462
999 463
999 464
999 465
999 466
999 467
999 468
999 469
999 470
999 471
999 472
999 473
999 474
999 475
999 476
999 477
999 478
999 479
999 480
999 481
999 482
999 483
999 484
999 485
999 486
999 487
999 488
999 489
999 490
999 491
999 492
999 493
999 494
999 495
999 496
999 497
999 498
999 499
999 500
999 501
999 502
999 503
999 504
999 505
999 506
999 507
999 508
999 509
999 510
999 511
999 512
999 513
999 514
999 515
999 516
999 517
999 518
999 519
999 520
999 521
999 522
999 523
999 524
999 525
999 526
999 527
999 528
999 529
999 530
999 531
999 532
999 533
999 534
999 535
999 536
999 537
999 538
999 539
999 540
999 541
999 542
999 543
999 544
999 545
999 546
999 547
999 548
999 549
999 550
999 551
999 552
999 553
999 554
999 555
999 556
999 557
999 558
999 559
999 560
999 561
999 562
999 563
999 564
999 565
999 566
999 567
999 568
999 569
999 570
999 571
999 572
999 573
999 574
999 575
999 576
999 577
999 578
999 579
999 580
999 581
999 582
999 583
999 584
999 585
999 586
999 587
999 588
999 589
999 590
999 591
999 592
999 593
999 594
999 595
999 596
999 597
999 598
999 599
999 600
999 601
999 602
999 603
999 604
999 605
999 606
999 607
999 608
999 609
999 610
999 611
999 612
999 613
999 614
999 615
999 616
999 617
999 618
999 619
999 620
999 621
999 622
999 623
999 624
999 625
999 626
999 627
999 628
999 629
999 630
999 631
999 632
999 633
999 634
999 635
999 636
999 637
999 638
999 639
999 640
999 641
999 642
999 643
999 644
999 645
999 646
999 647
999 648
999 649
999 650
999 651
999 652
999 653
999 654
999 655
999 656
999 657
999 658
999 659
999 660
999 661
999 662
999 663
999 664
999 665
999 666
999 667
999 668
999 669
999 670
999 671
999 672
999 673
999 674
999 675
999 676
999 677
999 678
999 679
999 680
999 681
999 682
999 683
999 684
999 685
999 686
999 687
999 688
999 689
999 690
999 691
999 692
999 693
999 694
999 695
999 696
999 697
999 698
999 699
999 700
999 701
999 702
999 703
999 704
999 705
999 706
999 707
999 708
999 709
999 710
999 711
999 712
999 713
999 714
999 715
999 716
999 717
999 718
999 719
999 720
999 721
999 722
999 723
999 724
999 725
999 726
999 727
999 728
999 729
999 730
999 731
999 732
999 733
999 734
999 735
999 736
999 737
999 738
999 739
999 740
999 741
999 742
999 743
999 744
999 745
999 746
999 747
999 748
999 749
999 750
999 751
999 752
999 753
999 754
999 755
999 756
999 757
999 758
999 759
999 760
999 761
999 762
999 763
999 764
999 765
999 766
999 767
999 768
999 769
999 770
999 771
999 772
999 773
999 774
999 775
999 776
999 777
999 778
999 779
999 780
999 781
999 782
999 783
999 784
999 785
999 786
999 787
999 788
999 789
999 790
999 791
999 792
999 793
999 794
999 795
999 796
999 797
999 798
999 799
999 800
999 801
999 802
999 803
999 804
999 805
999 806
999 807
999 808
999 809
999 810
999 811
999 812
999 813
999 814
999 815
999 816
999 817
999 818
999 819
999 820
999 821
999 822
999 823
999 824
999 825
999 826
999 827
999 828
999 829
999 830
999 831
999 832
999 833
999 834
999 835
999 836
999 837
999 838
999 839
999 840
999 841
999 842
999 843
999 844
999 845
999 846
999 847
999 848
999 849
999 850
999 851
999 852
999 853
999 854
999 855
999 856
999 857
999 858
999 859
999 860
999 861
999 862
999 863
999 864
999 865
999 866
999 867
999 868
999 869
999 870
999 871
999 872
999 873
999 874
999 875
999 876
999 877
999 878
999 879
999 880
999 881
999 882
999 883
999 884
999 885
999 886
999 887
999 888
999 889
999 890
999 891
999 892
999 893
999 894
999 895
999 896
999 897
999 898
999 899
999 900
999 901
999 902
999 903
999 904
999 905
999 906
999 907
999 908
999 909
999 910
999 911
999 912
999 913
999 914
999 915
999 916
999 917
999 918
999 919
999 920
999 921
999 922
999 923
999 924
999 925
999 926
999 927
999 928
999 929
999 930
999 931
999 932
999 933
999 934
999 935
999 936
999 937
999 938
999 939
999 940
999 941
999 942
999 943
999 944
999 945
999 946
999 947
999 948
999 949
999 950
999 951
999 952
999 953
999 954
999 955
999 956
999 957
999 958
999 959
999 960
999 961
999 962
999 963
999 964
999 965
999 966
999 967
999 968
999 969
999 970
999 971
999 972
999 973
999 974
999 975
999 976
999 977
999 978
999 979
999 980
999 981
999 982
999 983
999 984
999 985
999 986
999 987
999 988
999 989
999 990
999 991
999 992
999 993
999 994
999 995
999 996
999 997
999 998
999 999
999 1000
999 1001
999 1002
999 1003
999 1004
999 1005
999 1006
999 1007
999 1008
999 1009
999 1010
999 1011
999 1012
999 1013
999 1014
999 1015
999 1016
999 1017
999 1018
999 1019
999 1020
999 1021
999 1022
999 1023
999 1024
999 1025
999 1026
999 1027
999 1028
999 1029
999 1030
999 1031
999 1032
999 1033
999 1034
999 1035
999 1036
999 1037
999 1038
999 1039
999 1040
999 1041
999 1042
999 1043
999 1044
999 1045
999 1046
999 1047
999 1048
999 1049
999 1050
999 1051
999 1052
999 1053
999 1054
999 1055
999 1056
999 1057
999 1058
999 1059
999 1060
999 1061
999 1062
999 1063
999 1064
999 1065
999 1066
999 1067
999 1068
999 1069
999 1070
999 1071
999 1072
999 1073
999 1074
999 1075
999 1076
999 1077
999 1078
999 1079
999 1080
999 1081
999 1082
999 1083
999 1084
999 1085
999 1086
999 1087
999 1088
999 1089
999 1090
999 1091
999 1092
999 1093
999 1094
999 1095
999 1096
999 1097
999 1098
999 1099
999 1100
999 1101
999 1102
999 1103
999 1104
999 1105
999 1106
999 1107
999 1108
999 1109
999 1110
999 1111
999 1112
999 1113
999 1114
999 1115
999 1116
999 1117
999 1118
999 1119
999 1120
999 1121
999 1122
999 1123
999 1124
999 1125
999 1126
999 1127
999 1128
999 1129
999 1130
999 1131
999 1132
999 1133
999 1134
999 1135
999 1136
999 1137
999 1138
999 1139
999 1140
999 1141
999 1142
999 1143
999 1144
999 1145
999 1146
999 1147
999 1148
999 1149
999 1150
999 1151
999 1152
999 1153
999 1154
999 1155
999 1156
999 1157
999 1158
999 1159
999 1160
999 1161
999 1162
999 1163
999 1164
999 1165
999 1166
999 1167
999 1168
999 1169
999 1170
999 1171
999 1172
999 1173
999 1174
999 1175
999 1176
999 1177
999 1178
999 1179
999 1180
999 1181
999 1182
999 1183
999 1184
999 1185
999 1186
999 1187
999 1188
999 1189
999 1190
999 1191
999 1192
999 1193
999 1194
999 1195
999 1196
999 1197
999 1198
999 1199
999 1200
999 1201
999 1202
999 1203
999 1204
999 1205
999 1206
999 1207
999 1208
999 1209
999 1210
999 1211
999 1212
999 1213
999 1214
999 1215
999 1216
999 1217
999 1218
999 1219
999 1220
999 1221
999 1222
999 1223
999 1224
999 1225
999 1226
999 1227
999 1228
999 1229
999 1230
999 1231
999 1232
999 1233
999 1234
999 1235
999 1236
999 1237
999 1238
999 1239
999 1240
999 1241
999 1242
999 1243
999 1244
999 1245
999 1246
999 1247
999 1248
999 1249
999 1250
999 1251
999 1252
999 1253
999 1254
999 1255
999 1256
999 1257
999 1258
999 1259
999 1260
999 1261
999 1262
999 1263
999 1264
999 1265
999 1266
999 1267
999 1268
999 1269
999 1270
999 1271
999 1272
999 1273
999 1274
999 1275
999 1276
999 1277
999 1278
999 1279
999 1280
999 1281
999 1282
999 1283
999 1284
999 1285
999 1286
999 1287
999 1288
999 1289
999 1290
999 1291
999 1292
999 1293
999 1294
999 1295
999 1296
999 1297
999 1298
999 1299
999 1300
999 1301
999 1302
999 1303
999 1304
999 1305
999 1306
999 1307
999 1308
999 1309
999 1310
999 1311
999 1312
999 1313
999 1314
999 1315
999 1316
999 1317
999 1318
999 1319
999 1320
999 1321
999 1322
999 1323
999 1324
999 1325
999 1326
999 1327
999 1328
999 1329
999 1330
999 1331
999 1332
999 1333
999 1334
999 1335
999 1336
999 1337
999 1338
999 1339
999 1340
999 1341
999 1342

```

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

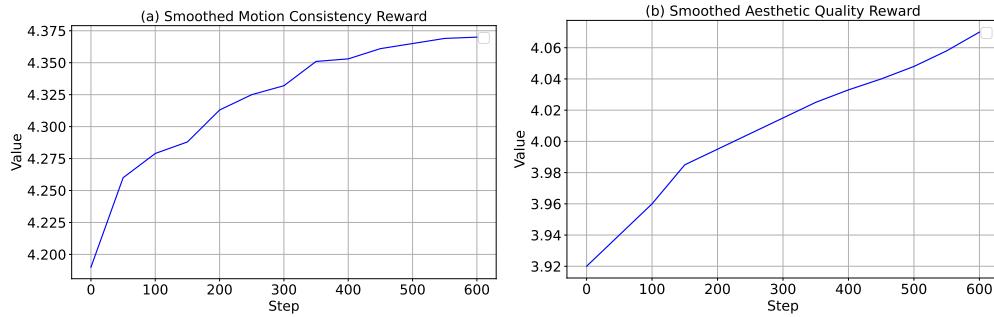


Figure 6: Smoothed Reward Curves for Motion Consistency and Aesthetic Quality

usually appealing outputs. The motion consistency reward rises steadily, indicating better alignment with the target trajectories, while the aesthetic reward demonstrates rapid gains in the early stages followed by a slower convergence, suggesting progressive refinement in visual quality. Together, these smoothed reward curves highlight the effectiveness of our reinforcement learning design in balancing motion control and perceptual quality.

D LLM USAGE STATEMENT

ChatGPT was employed solely for minor editorial assistance, such as improving grammar and readability. The research ideas, methodology, experiments, and analysis were entirely developed and conducted by the authors without the use of LLMs.